APPENDIX C
COPIES OF RELEVANT PRIOR ART REFERENCES

US005515488A

United States Patent [19]

Hoppe et al.

[11] Patent Number: 5,515,488

[45] Date of Patent: May 7, 1996

[54] **METHOD AND APPARATUS FOR CONCURRENT GRAPHICAL VISUALIZATION OF A DATABASE SEARCH AND ITS SEARCH HISTORY**

[75] Inventors: **Eric A. Hoppe**, San Jose; **Ramana B. Rao**; **Jock Mackinlay**, Palo Alto, all of Calif.

[73] Assignee: **Xerox Corporation**, Stamford, Conn.

[21] Appl. No.: **297,996**

[22] Filed: **Aug. 30, 1994**

[51] **Int. Cl.**[6] **G06F 17/30**

[52] **U.S. Cl.** **395/140**; 395/155; 395/160; 395/161

[58] **Field of Search** 395/155–161, 395/140, 600, 650; 364/419.01, 419.07, 419.19

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,649,499	3/1987	Sutton et al.	364/518
5,021,976	6/1991	Wexelblat et al.	364/521
5,065,347	11/1991	Pajak et al.	395/159
5,333,254	7/1994	Robertson	395/155
5,339,390	8/1994	Robertson et al.	395/157
5,355,473	10/1994	Au	395/600

OTHER PUBLICATIONS

"XSoft Brings Document Management to PCs," The Seybold Report on Desktop Publishing, Apr. 4, 1994. vol. 8, No. 8, pp. 29–30.

Vizard, M., "Document Manager Taps Data Visualizer," PC Week, Apr. 18, 1994, pp. 63–64.

Furnas, G. W., "Generalized Fisheye Views," CHI '86 Proceedings, ACM, Apr. 1986, pp. 16–23.

Spoerri, A., "InfoCrystal: A visual tool for information retrieval," Center for Educational Computing Initiatives, M.I.T., Cambridge, Maryland, IEEE 1993, pp. 150–157.

Fairchild, K. M., Poltrock, S. E., and Furnas, G. W., "SemNet: Three-Dimensional Graphic Representation of Large Knowledge Bases," in Guindon, R., Ed., Cognitive Science and its Application for Human Computer Interaction, Lawrence Erlbaum, Hillsdale, New Jersey, 1988, pp. 201–233.

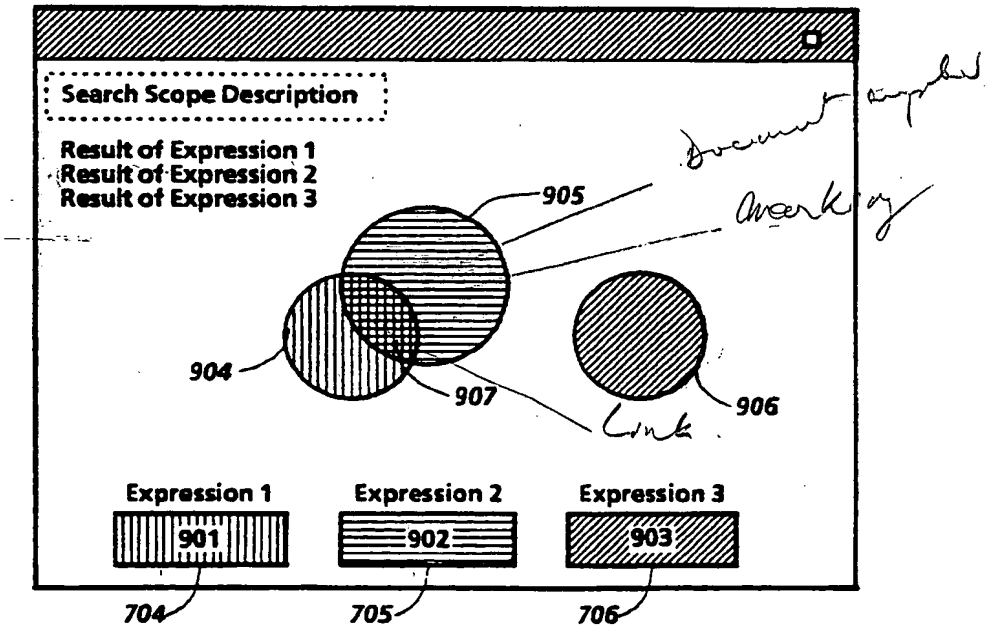
Primary Examiner—Almis R. Jankus

Attorney, Agent, or Firm—Richard B. Domingo

[57] **ABSTRACT**

A computer controlled display system providing for graphical representation of a query to a database and creation and traversal through a search history. A database search is typically performed by a sequence of narrowing queries. Each narrowing query is performed in a query window. A query window is comprised of an input area for entering query expressions, an query results display area, an indicator of a search scope associated with the query window and a history indicator area. A suitable information visualization technique is used to graphically display the search results in the query results display area. From these visualizations, new search scopes and query windows are created. A search path comprising the query windows for the current search path are displayed at any instant of time of the search. A history mechanism provides for ready traversal through the search history.

18 Claims, 8 Drawing Sheets



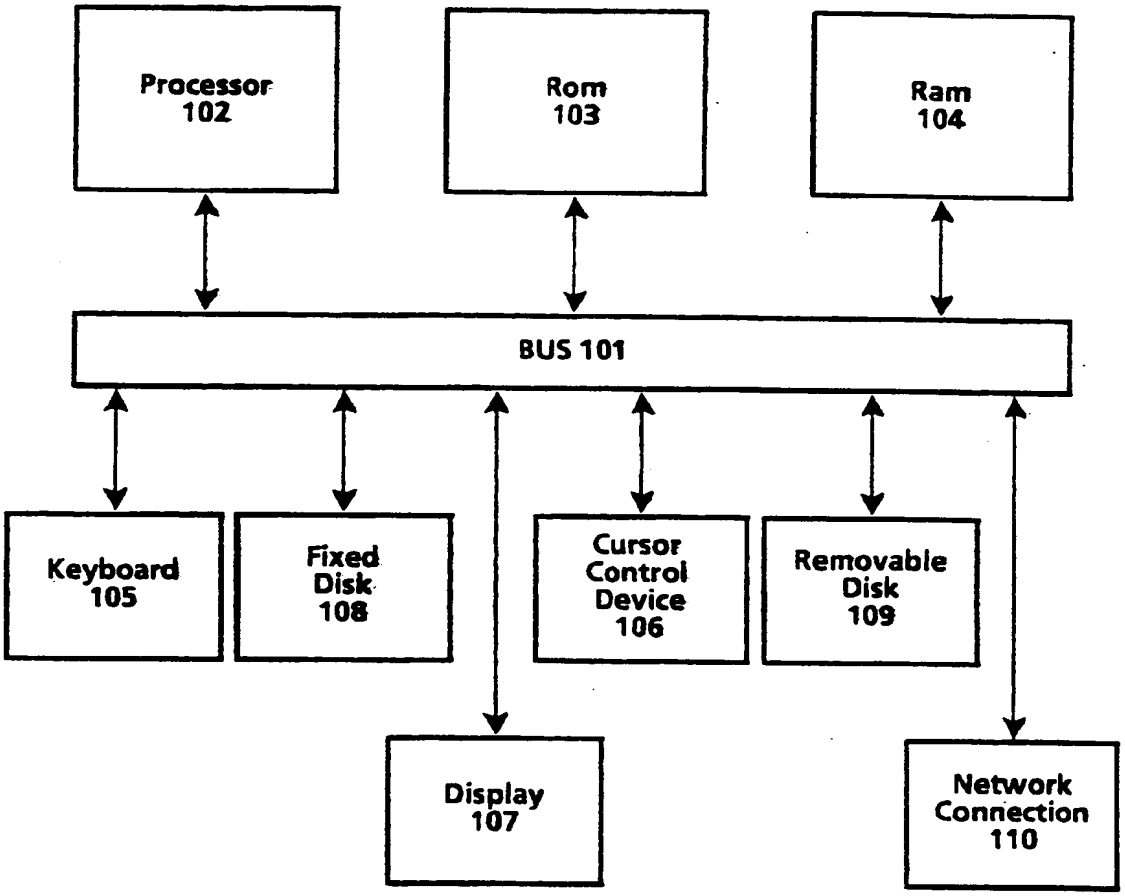

Processor
102
Rom
103
Ram
104
BUS 101
Keyboard
105
Fixed
Disk
108
Cursor
Control
Device
106
Removable
Disk
109
Display
107
Network
Connection
110

Fig. 1

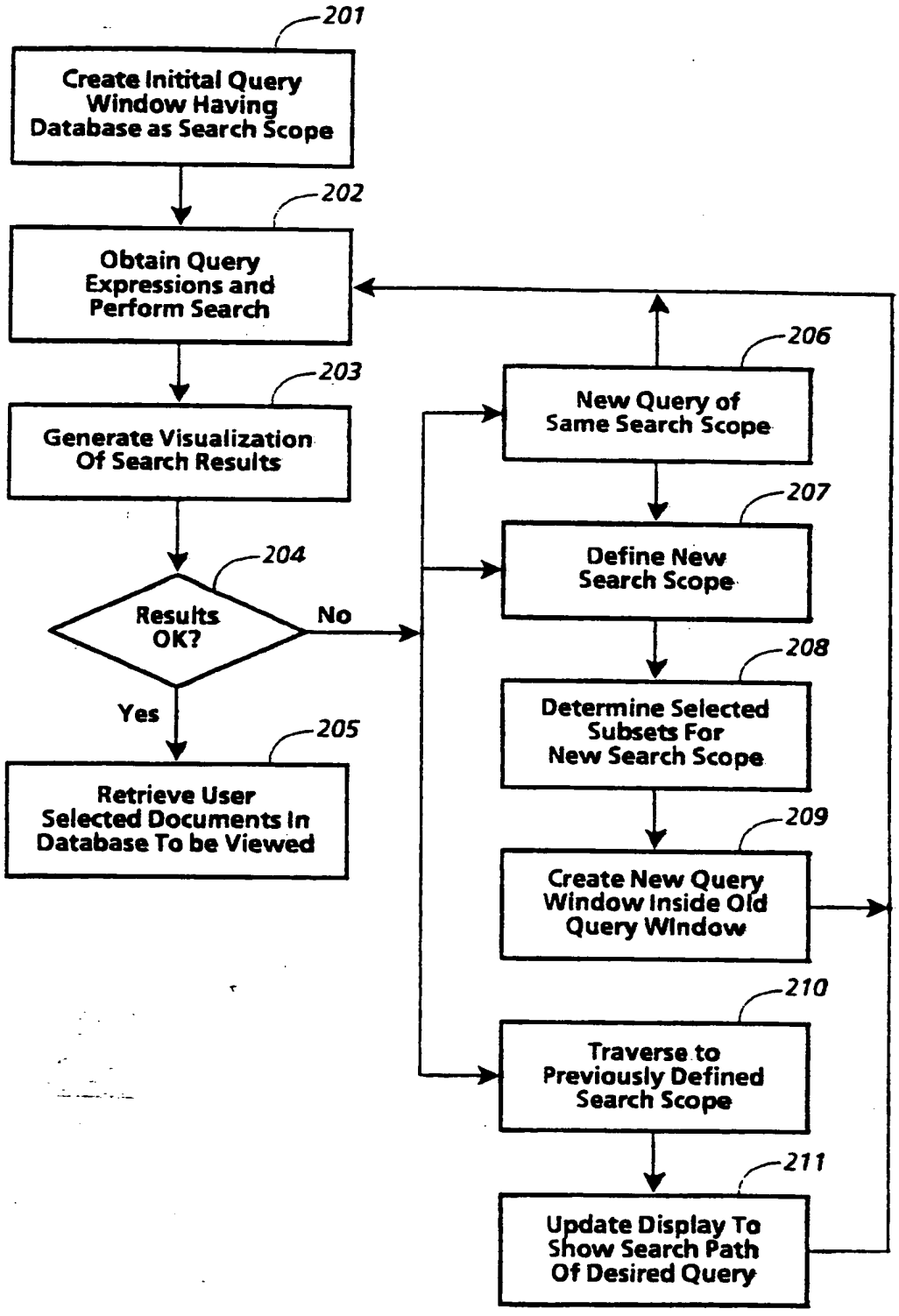

201
Create Initital Query Window Having Database as Search Scope
202
Obtain Query Expressions and Perform Search
203
Generate Visualization Of Search Results
204
Results OK?
No
Yes
205
Retrieve User Selected Documents In Database To be Viewed
206
New Query of Same Search Scope
207
Define New Search Scope
208
Determine Selected Subsets For New Search Scope
209
Create New Query Window Inside Old Query Window
210
Traverse to Previously Defined Search Scope
211
Update Display To Show Search Path Of Desired Query

Fig. 2

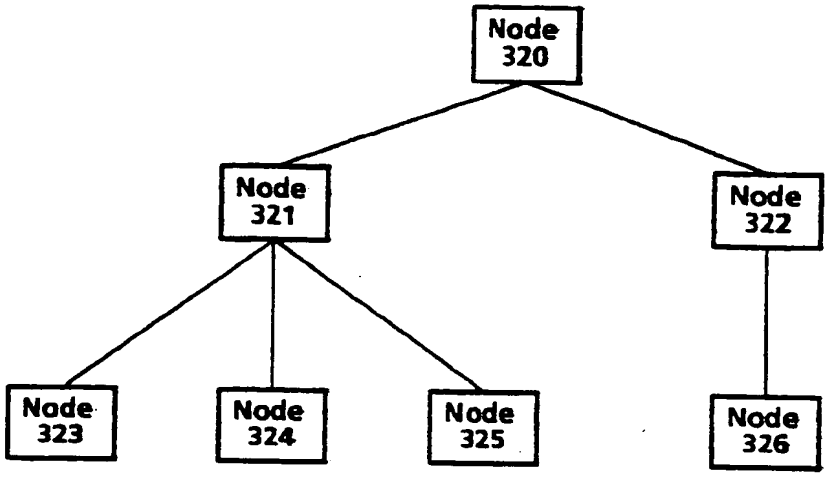

Node 320
Node 321
Node 322
Node 323
Node 324
Node 325
Node 326

Fig. 3

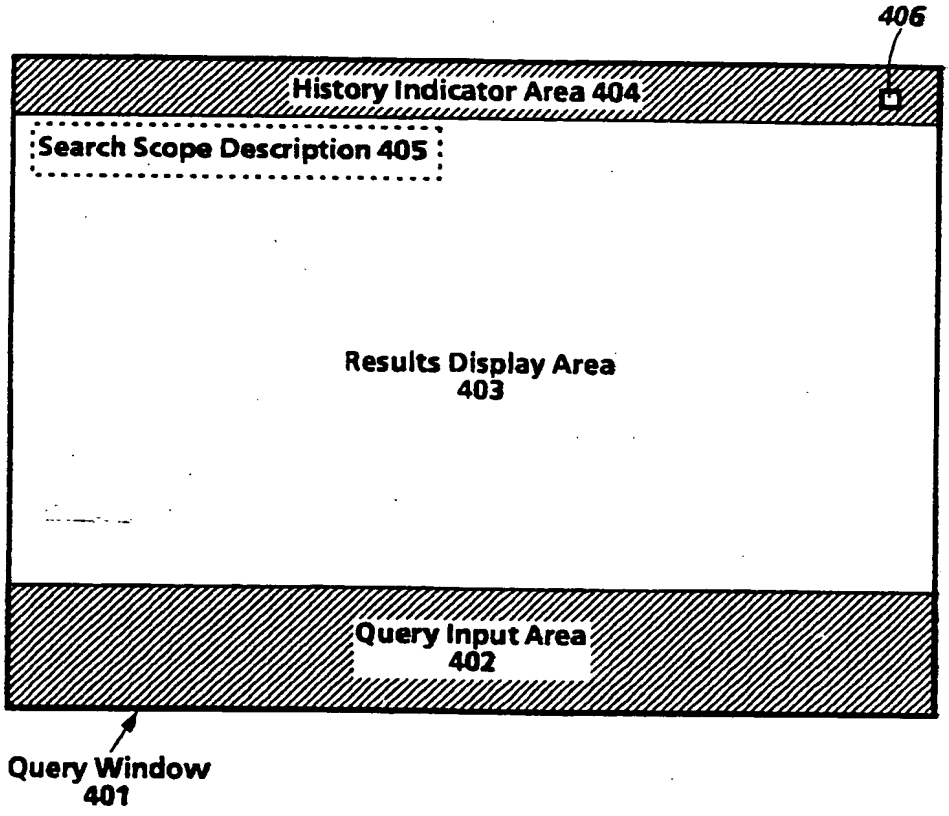

406
History Indicator Area 404
Search Scope Description 405
Results Display Area 403
Query Input Area 402
Query Window 401

Fig. 4

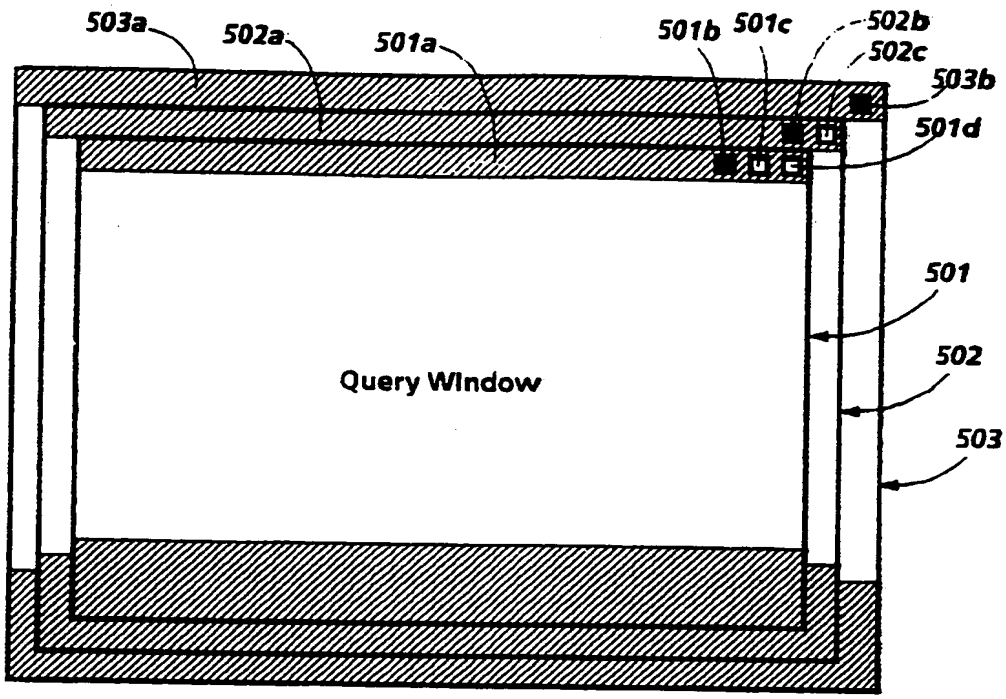

503a
502a
501a
501b
501c
502b
502c
503b
501d
501
502
503
Query Window

Fig.5a

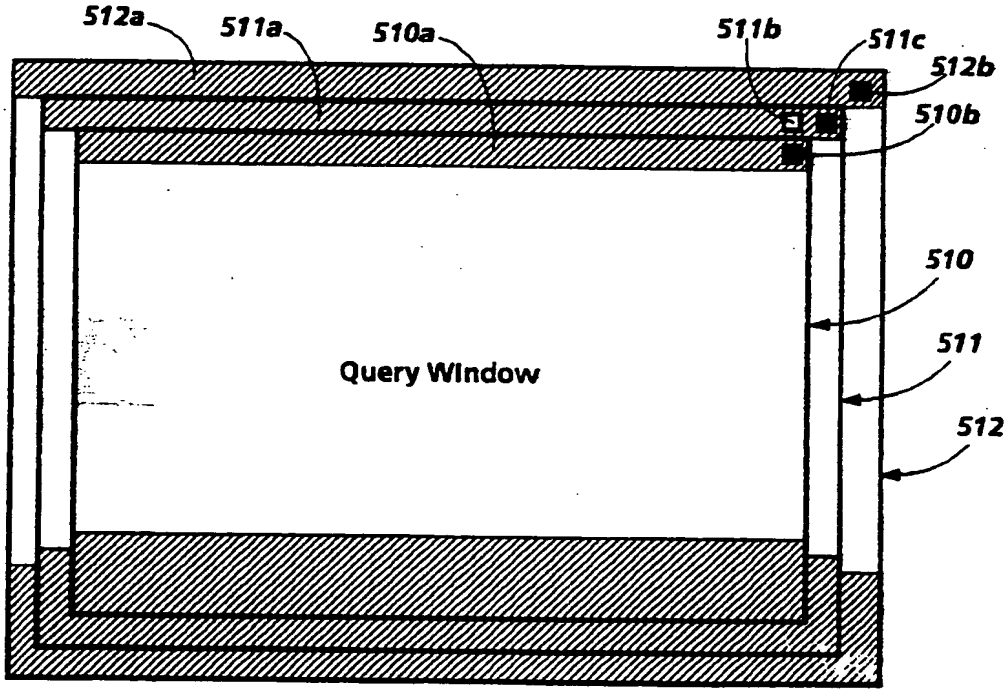

512a
511a
510a
511b
511c
512b
510b
510
511
512
Query Window

Fig.5b

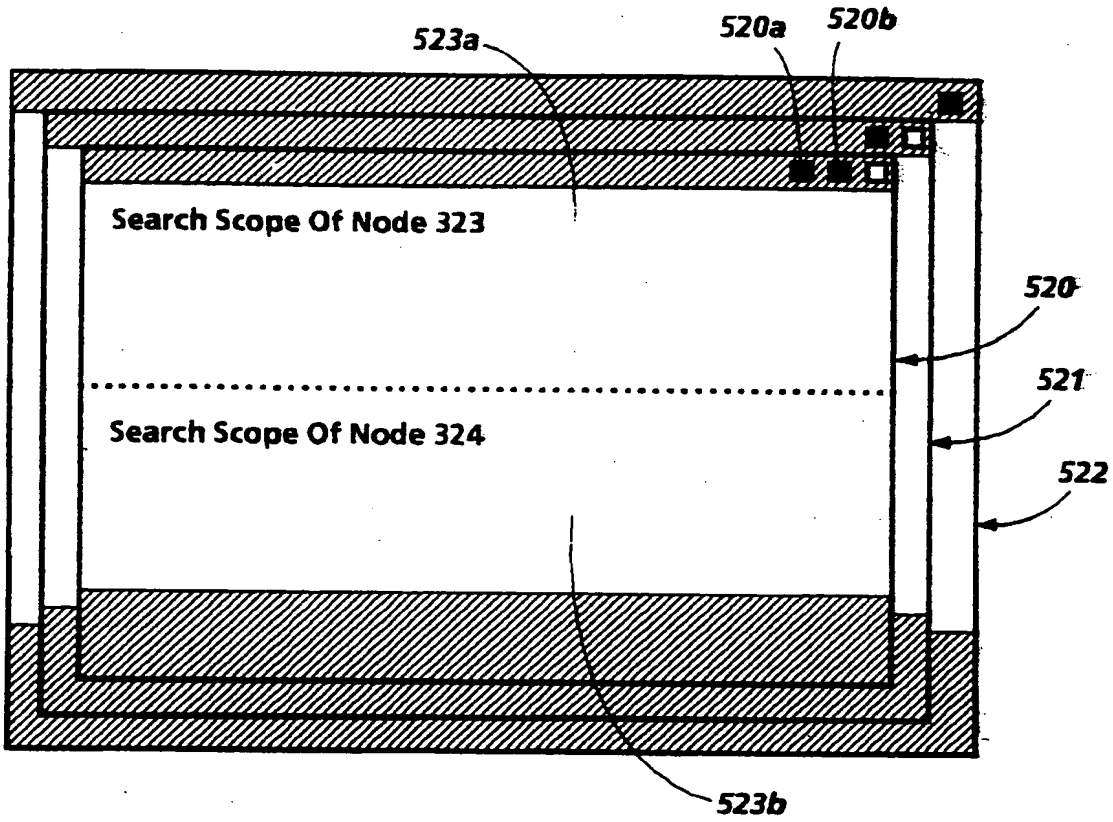

523a
520a
520b
Search Scope Of Node 323
520
521
Search Scope Of Node 324
522
523b

Fig.5c

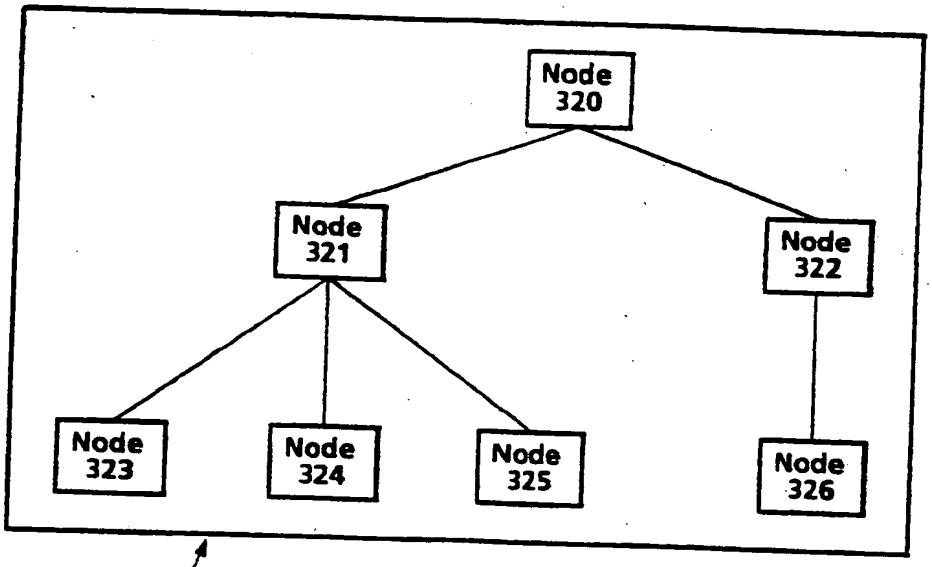
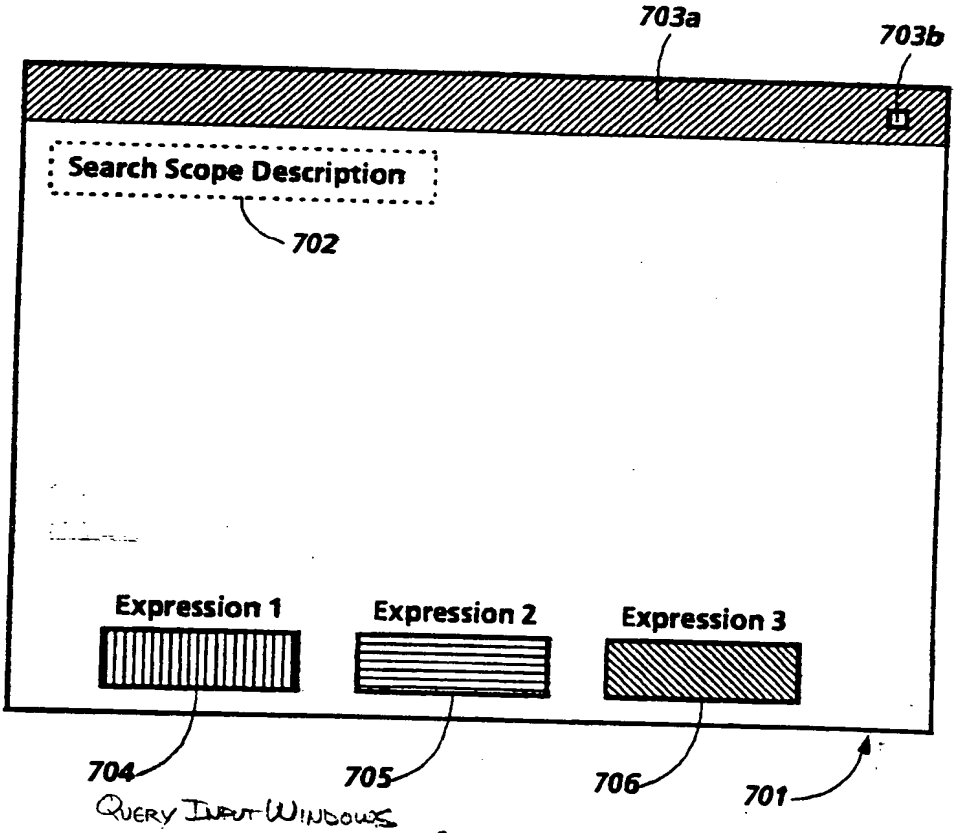


Fig. 6



Query Input Windows

Fig. 7

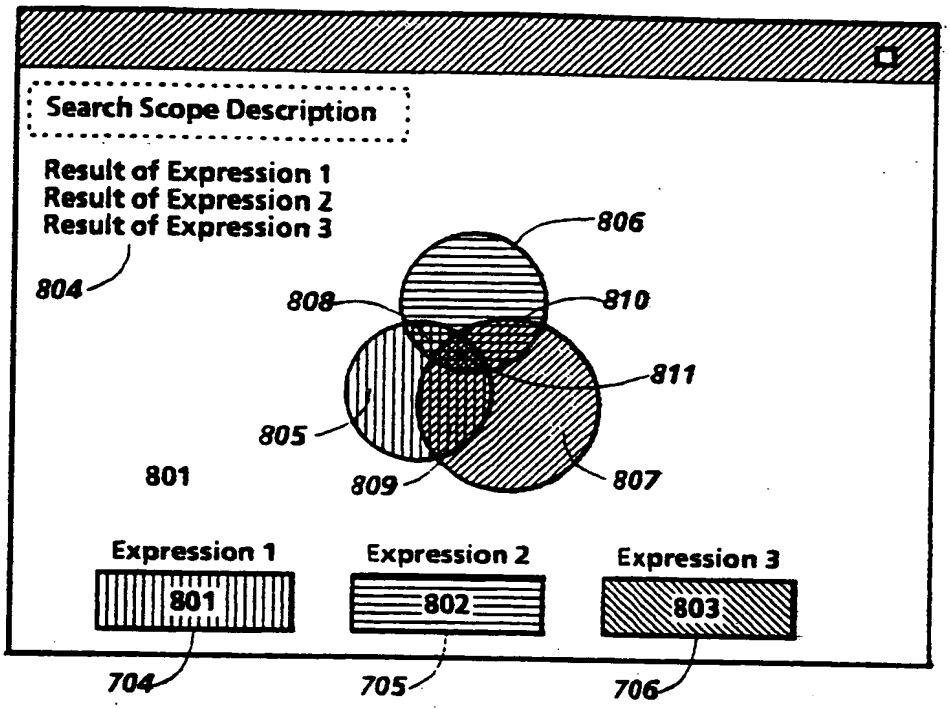

Search Scope Description
Result of Expression 1
Result of Expression 2
Result of Expression 3
804
806
808
810
811
805
801
809
807
Expression 1
801
Expression 2
802
Expression 3
803
704
705
706

Fig. 8

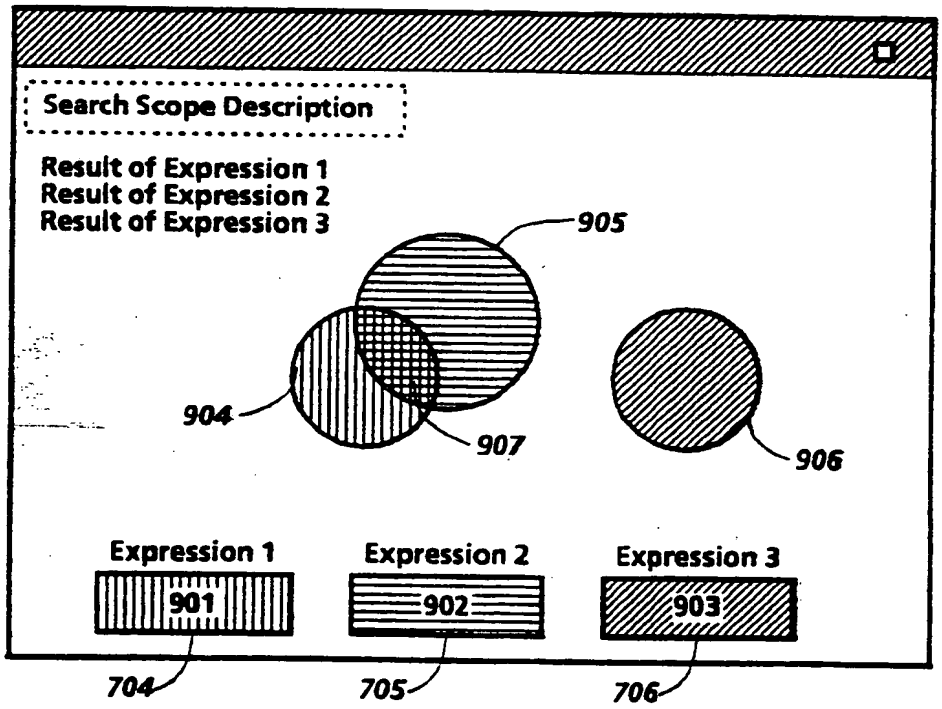

Search Scope Description
Result of Expression 1
Result of Expression 2
Result of Expression 3
905
904
907
906
Expression 1
901
Expression 2
902
Expression 3
903
704
705
706

Fig. 9

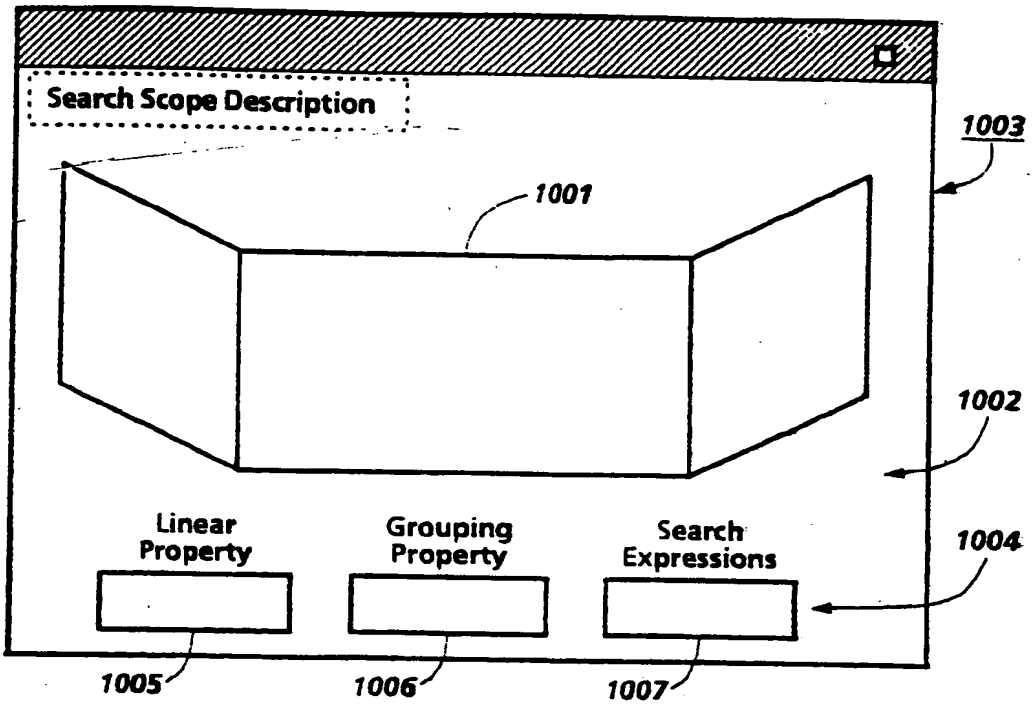

Search Scope Description
1003
1001
1002
Linear Property
Grouping Property
Search Expressions
1004
1005
1006
1007

Fig. 10

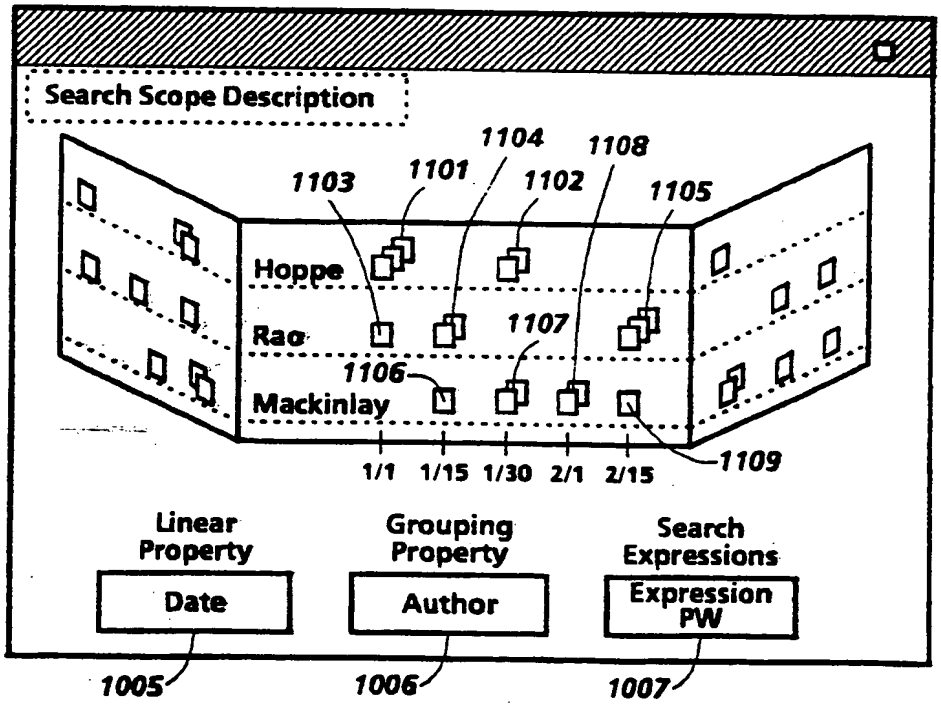

Search Scope Description
1104
1108
1103
1101
1102
1105
Hoppe
Rao
1107
1106
Mackinlay
1109
1/1
1/15
1/30
2/1
2/15
Linear Property
Grouping Property
Search Expressions
Date
Author
Expression PW
1005
1006
1007

Fig. 11

METHOD AND APPARATUS FOR CONCURRENT GRAPHICAL VISUALIZATION OF A DATABASE SEARCH AND ITS SEARCH HISTORY

FIELD OF THE INVENTION

The present invention relates generally to the field of information visualization, and more particularly, graphical visualization of a database search.

BACKGROUND OF THE INVENTION

More and more information is being made available to computer system users via various mediums such as CD-ROMs, on-line databases and the like (collectively referred to as databases). A query to a database typically requires a complex textual specification based on keywords and logical relationships between sets of information. In most instances, the query returns only the results. Often, the results are not useful either because the results are much larger than that which can be easily visualized and manipulated, or because the result is unexpectedly empty.

When performing a search, it is typical that a search strategy will be used in order to find the desired information. Most search strategies are premised on attaining a reasonable number of items that satisfy a search criteria. Typically, a query is comprised of keywords (i.e. search terms) connected together via Logical and/or Proximity Operators. Logical Operators are used to include or exclude items in a set whereas proximity operators are used to identify items having keywords that are a predetermined distance apart (such as within 10 word or that are adjacent). Once a query is made and executed, a list of items satisfying the criteria of the query is presented to the user. The user can then either view one or more items in the list, or if the list is large, modify the search to reduce the number of items in the list.

One prior art system, the LEXIS Information retrieval system, allows queries to be performed according to various levels. Each subsequent level contains a subset of the results of the immediately prior level, based on user provided search criteria. The LEXIS system provides text based feedback which indicates the number of items found which satisfy the search criteria. The user then has various options to view the list of items found (e.g. full text, keyword in context, segments or as a list of citations.)

A second prior art system is the DIALOG information retrieval system. In DIALOG, query results can be structured so that feedback is provided as to the number of items found which satisfy each keyword. Queries may also be combined to create new queries. However, the user must track the queries made in order to make effective use of these facilities.

When performing searches, it may also be desirable to be able to restart searches at a point in the middle of a search path. In the aforementioned LEXIS System, this is accomplished by specifying and modifying a prior search level. This has the drawback in that it entirely replaces the prior search level and all search level below the level modified. In the Dialog system this can be done, but is left to the user to map out the query history according to the taken search sequence. No mechanism is provided to the user to accommodate this. Thus, it would be desirable to have a system that is capable of creating a search history through which a user may restart searches at designated points without destroying the results of any prior searching.

Further materials relevant to present invention include:

EP 0 535 986 A2, entitled "Method of Operating A Processor", Robertson, which is assigned to the assignee of the present invention describes a method for centering a selected node of a node link structure along a centering line. The nodes are in rows, and each row extends across a centering line with links between nodes in adjacent rows. When a user requests a centering operation for an indicated node, a sequence of images is presented, each including a row that appears to be a continuation of the row with the indicated node and that includes a continued indicated node that appears to be a continuation of the indicated node. The rows appear to be shifted, bringing the continued indicated nodes toward the centering line, until a final shift locks the continued indicated node into position at the centering line. The positions of the indicated node and a subset of the continued indicated nodes together can define an asymptotic path that begins at the position of the indicated node and approaches the center line asymptotically until the final shift occurs. The displacements between positions can follow a logarithmic function, with each displacement being a proportion of the distance from the preceding position to the centering line. Each node can be rectangular, and the nodes in each row can be separated by equal offsets to provide compact rows. Each node can be a selectable unit, so that the user can request a centering operation by selecting a node, such as with a mouse click.

EP 0447 095A, Robertson, et al., entitled "Workspace Display", which is assigned to the assignee of the present invention discloses a processor which presents a sequence of images of a workspace that is stretched to enable the user to view a part of a workspace in greater detail. The workspace includes a middle section and two peripheral sections that meet the middle section on opposite edges. Each of the sections appears to be a rectangular two-dimensional surface and they are perceptible in three dimensions. When the user is viewing the middle section as if it were parallel to the display screen surface, each peripheral section appears to extend away from the user at an angle from the edge of the middle section so that the peripheral sections occupy relatively little of the screen. When the user requests stretching, the middle section is stretched and the peripheral sections are compressed to accommodate the stretching. When the user requests destretching, the middle section is destretched and the peripheral sections are decompressed accordingly.

Furnas, G. W., "Generalized Fisheye Views," CHI '86 Proceedings, ACM, April 1986, pp. 16–23, describes fisheye views that provide a balance of local detail and global context. Section 1 discusses fisheye lenses that show places nearby in great detail while showing the whole world, showing remote regions in successively less detail; a caricature is the poster of the "New Yorker's View of the United States." Section 3 describes a degree of interest/DOI) function that assigns to each point in a structure, a number telling how interested the user is in seeing that point, given the current task. A display can then be made by showing the most interesting points, as indicated by the DOI function. The fisheye view can achieve, for example, a logarithmically compressed display of a tree, as illustrated by FIG. 4 of Furnas for a tree structured text file. Section 4 also describes fisheye views for botanical taxonomies, legal codes, text outlines, a decisions tree, a telephone area code directory, a corporate directory, and UNIX file hierarchy listings. Section 5 indicates that a display-relevant notion of a priori importance can be defined for lists, trees, acylic directed graphs, general graphs, and Euclidean spaces, unlike the geographical example which inspired the metaphor of the

"New Yorker's View," the underlying structures need not be spatial, nor need the output be graphic. FIG. 6 of Furnas shows a fisheye calendar.

Spoerri, Anselm, "InfoCrystal: A visual tool for information retrieval", MIT-CETI-TR 93-3, describes with reference to a FIG. 1, how to transform a Venn diagram into an iconic display which represents all possible Boolean queries involving its inputs in a normal form. The Venn diagram is first exploded into its disjoint subsets. The subsets are then represented by icons whose shapes reflect the number of criteria satisfied by their contents (also called the rank of a subset.) Finally, the subset icons are surround by a border area that contains criterion icons that represent the original sets. Visual coding principles that are incorporated include (1) shape coding to indicate the number of criteria that the contents associated with an interior icon satisfy, (2) proximity coding to indicate that the closer an interior icon is located to a criterion icon, the more likely it is that the icon's contents are related to it, (3) rank coding to indicate how many criteria are satisfied, (4) color or texture coding to indicate which particular criteria are satisfied by the icon's contents, (5) orientation coding so that the sides of an icon are positioned so that their sides face the criteria they satisfy, and (6) size or brightness & saturation coding to indicate the number of elements represented by an icon. Section 2.2 describes a Visual Query Language wherein the output of an InfoCrystal is defined as a set of selected interior icons. FIG. 3 illustrates how the InfoCrystals can be "chained together" to form a hierarchical query structure.

SUMMARY OF THE INVENTION

A computer controlled display system providing for graphical representation of a query to a database and creation and traversal through a search history is disclosed. In the present invention, the results of a query to a database are graphically displayed in a query window using a suitable information visualization technique. The information visualization causes the display of the query results as one or more disjoint and selectable graphical regions relative to a search scope (e.g. a Venn diagram situated on a plane). The query window is further comprised of an input area for entering query expressions, an indicator of a search scope associated with the query window and a history indicator area. The history indicator area contains icons identifying siblings within a search level. The query windows in a particular search path are displayed as concentrically nested to provide a visual cue as to the relationship of the query windows. Where a query window is in the nesting indicates it's level in the search history. The nesting further provides for easy traversal through that search path can be accomplished in a point and click fashion. New query windows are created by definition of a new search scope based on search results.

The present invention further provides a search history mechanism for facilitating traversal through the search history. One aspect of the search history mechanism is embodied in the history indicator areas in each of the query windows. The alignment of the query windows and their corresponding indicators reveal a branch of the search history. Traversal to particular points in the path is enabled by clicking on the an icon associated with the desired query window (or search scope). A second aspect of the search history mechanism is the provision of a history windows for displaying the search history in a tree format.

DESCRIPTION OF THE DRAWINGS

FIG. 1 is a block diagram of a computer based system upon which the currently preferred embodiment of the present invention may be implemented.

FIG. 2 is a flowchart illustrating the basic steps in a database search which result in the creation of query windows and a search history tree, as may be performed in the currently preferred embodiment of the present invention.

FIG. 3 is an example of a search history tree as may be created in the present invention.

FIG. 4 illustrates a query window as may be utilized by the currently preferred embodiment of the present invention.

FIG. *5a* illustrates a first configuration of nested query windows illustrating a first search path of the search history tree of FIG. 3, as may be utilized by the currently preferred embodiment of the present invention.

FIG. *5b* illustrates a second configuration of nested query windows illustrating a second search path of the search history tree of FIG. 3, as may be utilized by the currently preferred embodiment of the present invention.

FIG. *5c* illustrates a third configuration of nested query windows illustrating a third search path and said first search path of the search history of FIG. 3 displayed in the same query window, as may be utilized by the currently preferred embodiment of the present invention.

FIG. 6 illustrates a history window displaying the search history tree of FIG. 3, as may be utilized by the currently preferred embodiment of the present invention.

FIG. 7 illustrates a venn diagram visualization for a query window as may be used the currently preferred embodiment of the present invention.

FIG. 8 illustrates the query window of FIG. 7 after an executed query showing a graphical visualization of the query results using a venn diagram

FIG. 9 illustrates an update of the query window of FIG. 7 after an executed query showing a graphical visualization of the query results using a venn diagram where one expression of the query has no elements in common with the other expressions.

FIG. 10 illustrates a perspective wall implementation of a query window in the currently preferred embodiment of the present invention.

FIG. 11 illustrates the query window of FIG. 10 after an executed query showing a graphical visualization of the query results using a perspective wall.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A computer controlled display system for graphically displaying the results of a query to a database is disclosed. In the following description numerous specific details are set forth, such as the operational aspects of a database, in order to provide a thorough understanding of the present invention. It would be apparent, however, to one skilled in the art to practice the invention without such specific details. In other instances, specific implementation details, such as software coding techniques for creating graphical objects, have not been shown in detail in order not to unnecessarily obscure the present invention.

The term database as used herein refers to any body of information that is accessible via a computer based system. The body of information would typically be a database located on a storage medium directly connected to the

computer based system (e.g. on a CD-ROM) or accessible via a network (e.g. an on-line information source). Alternatively, the body of information could be a collection of documents or document parts managed by a document management system. In any event, such databases can be characterized as having three primary parts: the information or data itself, a retrieval/updating part and a user interface part. The retrieval/updating part enables access to the information for retrieval, editing, or addition of information. The user interface part is the mechanism by which a user interacts with the database to search for and obtain information. It is this user interface part to which the present invention is directed.

As used herein, the term search refers to the steps performed for retrieval of information from a database. The term document refers to the specific items of information contained in the database. Documents include textual, audio or visual works. The term search scope refers to the set of information in the database which may be retrieved at any instant during the search. As will become apparent in the foregoing description, a search scope is created through selection of subsets of the results of a query. The search scope is narrowed as the various queries in the search are performed. The term query refers to a set of parameters provided for executing a step in a search. This set of parameters is typically in the form of one or more expressions. Expressions are predicates such as keywords or sets of keywords, dates, numbers and other data types, as well as various combination thereof, combined with logical or proximity operators which define the documents of interest.

Overview of a Computer Based System In the Currently Preferred Embodiment of the Present Invention

The computer based system on which the currently preferred embodiment of the present invention may be implemented is described with reference to FIG. 1. Referring to FIG. 1, the computer based system is comprised of a plurality of components coupled via a bus **101**. The bus **101** illustrated here is simplified in order not to obscure the present invention. The bus **101** may consist of a plurality of parallel buses (e.g. address, data and status buses) as well as a hierarchy of buses (e.g. a processor bus, a local bus and an I/O bus). In any event, the computer system is further comprised of a processor **102** for executing instructions provided via bus **101** from either static Read Only Memory (ROM) **103** or dynamic Random Access Memory (RAM) **104**. The processor **102**, ROM **103** and RAM **104** may be discrete components or a single integrated device such as an Application Specification Integrated Circuit (ASIC).

It should further be noted that the processor **102** is used to execute instructions coded in a suitable programming language for creating the graphical data used to create the query and history windows and the various visualizations described herein. The processor **102** would also process the queries associated with a database search. Moreover, such instructions would be used for causing the steps outlined in FIG. 2 to be performed.

Also coupled to the bus **101** are a keyboard **105** for entering alphanumeric input, a cursor control device **106** for manipulating a cursor, a display **107** for displaying visual output, and a fixed disk **108** for storing data (e.g. the database). The fixed disk **108** may be a magnetic or optical disk. Further coupled to the bus **101** is a removable disk **109** and a network connection **110**. The removable disk **109** may be a floppy disk drive, or an optical disk drive such as a CD-ROM drive which itself may be a database). The network connection **110** represents either a local area network coupling or a public network coupling. In any event, the network connection **110** provides for coupling to databases residing on the network.

The representation of the results of a query will be displayed on display **107** and the user will interact with the computer controlled based system through a combination of the keyboard **105** and the cursor control device **106**.

While the preferred embodiment of the present invention is embodied on a computer based system, the present invention could be practiced on any computer controlled display system, such as a fixed function terminal. The currently preferred embodiment of the present invention is implemented on a computer controlled display system having a Graphical User Interface (GUI) which allows multiple concurrent "windows" to be in operation (e.g. one of the family of Macintosh Computers available from Apple Computer, Inc. of Cupertino Calif.). A "window" refers to a visual representation of an executing task. Windows and operation thereof is well known in the art, so no further discussion of windows or their operation is deemed necessary. Such a GUI will also support operations such as "point and click". A "point and click" operation is one where a cursor on a display screen is positioned over a desired portion of the display, such as an icon, using a cursor control device such as a mouse or trackball. Once the cursor is appropriately positioned, a button/switch associated with the cursor control device is quickly depressed and released. This creates an electrical signal which causes a predetermined operation to occur. Other operations may require a "double click" where the button/switch is depressed and released rapidly, twice in succession.

Operational Flow and Creation of a Search History

FIG. 2 is a flowchart illustrating the basic steps of performing a search and the resulting creation of the history tree structure in the currently preferred embodiment. Accordingly, certain steps, such as selection of the desired visualization for the search results, are not described. Referring to FIG. 2, a query window is created having the entire database as a search scope, step **201**. Referring briefly to FIG. 3, this is represented as root node **320**. Referring back to FIG. 2, a search is then performed based on one or more search expressions, step **202**. The search expressions provided would be according to the search rules of the particular data retrieval mechanism associated with the database. For example, if the database was based on a relational database, the search expressions would be based on relationships between elements of known categories of data. The query will be processed, the desired visualization determined and a graphical visualization of the search results is displayed in the query window, step **203**. As the present invention may incorporate various visualization techniques, the user will have selected the desired visualization either before or after execution of the query. Alternatively, the visualization may be automatically selected based on the query. In any event, a user must then determine if they are satisfied with the search results (i.e. the results are o.k.), step **204**. If they are satisfied, the search to this point is completed and the user would select documents which would be retrieved for viewing, step **205**. If they are not satisfied, the user must then formulate a subsequent search strategy. Two options are to formulate a new query with the same search scope, step **206**, create a new search scope, step **207** or to traverse back to a

prior search scope, step **210**. For the option of formulating a new query to the same search scope, the query is obtained and the search performed per step **202**. For the option of defining a new search scope, the user selects the new search scope from the results of the last executed query. As will be explained in greater detail below, this will typically be done by a point and click operation on the visualization of the search results within the query window. The subsets of the search results are determined, step **208** and a new query window is created inside the previous query window, step **209**. This creating within a prior query window causes a "concentric nesting" of the query windows to provide a visual cue as to the relationship of the windows. Once the new search scope is created, queries are executed per step **202**. Referring briefly to FIG. **3**, this step is performed for creation for all the nodes except the root node **320**.

If the user decides to traverse back to a previously defined search scope per step **210**, the user must select the desired search scope. As will become apparent in the description below, the present invention provides two ways to accomplish this, via the history indicators in the query window of the concurrently displayed query windows or via the history structure displayed in the history window. In any event, once the desired search scope is selected, the display is updated to present the query windows corresponding to the path of the selected search scope, step **211**. New queries would then be executed, per step **202**, based on the selected search scope.

Referring back to FIG. **3**, it is presumed that the steps described in FIG. **2** are used to add the nodes **321–326**. It should be noted that the nodes **321** and **322** are on the same level and have the same parent node. The nodes **323–326** are on the same level but the nodes **323–325** have a different parent than node **326**. Thus, nodes **323–325** are said to be on different search paths than node **326**. As will be described below, when displaying a search path, the query window for a node for each level in the path is displayed.

Query Windows

The structure of a query window in the currently preferred embodiment is illustrated with respect to FIG. **4**. The term query window is meant to refer to the elements of the visual interface and does not limit the spirit and scope of the present invention. Examples of different implementations of a query window are provided in the description of the visual representation of query results provided below. Referring to FIG. **4**, a query window **401** has a query input area **402**, a results display area **403** and a history indicator area **404**. The query input area **402** is where a user inputs a query. As described above a query is comprised of one or more expressions. The composition of the query input area will depend on the type of visualization desired.

The results display area **403** will display a graphical visualization of the search results according to a selected information visualization technique. Visualizations such as venn diagrams, a perspective wall, a hierarchical representation, lists, or tables, may be utilized. The key criteria for a graphical visualization is that the results are presented as selectable disjoint subsets. The manner in which selection occurs will depend on the graphical visualization used. Examples of such visualizations and a corresponding selection technique are described in greater detail below.

Also present in the results display area **403** is a search scope description **405** for the query window. The search scope description **405** is typically a textual description of the search scope of the query window (e.g. a logical organization of the expressions used to achieve the search scope.)

The history indicator area **404** provides a means by which a user may visually determine a location within a hierarchical search path. The history indicator will contain a number of indicators representing different created search paths at the same search level. The indicators used may be an icon or text symbol(s). The indicator(s) representing the search scope being displayed (and the search path) are highlighted. Here, an icon **406** (a box) is displayed to indicate that there is one search scope defined at this level. Multiple icons may be present. Each icon in the history indicator area **404** corresponds to a search scope.

Variations of the placement of the described areas are within the scope of the present invention. For example, the history indicator area may run down a vertical side of the query window. This may be desirable if the history structure used in a history window was displayed with a horizontal orientation (rather than vertical orientation of FIG. **3**.) In such a case, the levels of the tree would naturally be in columns rather than rows. As a result, it would be more consistent with the history structure to have the icons displayed with a vertical orientation. Moreover, each of the various areas may be implemented as separate "windows". This would enable flexibility in the display of the query window. For example, it may be desirable at some point to remove a query input area when further query input is not needed.

As noted above, during the course of a database search, query windows corresponding to a direct search path are displayed in a nested concentric fashion. This is illustrated with reference to FIGS. ***5a*** and ***5b***. FIGS. ***5a*** and ***5b*** illustrate two search paths taken within the history structure of FIG. **3**. Referring to FIG. ***5a*** a plurality of query windows are displayed. There is a single query window displayed for each level in the search path. Only the query windows in the direct search path are displayed. However, the history indicator area is used to indicate the number of "sibling" nodes along search path that are on the same level. In FIG. ***5a*** three query windows are displayed. Query window **501** corresponds to the search scope of node **323**, query window **502** corresponds to the search scope of node **321** and query window **503** corresponds to the search scope of node **320**. For query window **501** there are three nodes in the search path at that level. This is indicated by the three boxes/ ***501b–501d***) in the history indicator area ***501a*** of query window **501**. Note that the box ***501b*** is highlighted to indicate the direct search path at this level of the tree. Moreover at this point it indicates where this query window is in the history structure. Similarly with respect to query window **502**, history indicator area ***502a*** is comprised of boxes ***502b–502c***. The box ***502b*** is highlighted to indicate the direct path for the search scope of query window **501**.

Referring now to FIG. ***5b***, again three windows **510–512** are displayed and correspond to the three levels of the tree structure. Query window **510** corresponds to the search scope of node **326**, query window **511** corresponds to the search scope of node **322** and query window **512** corresponds to the search scope of node **326**. For query window **510** there is only one node at that level of the search path so there is only one box ***510b*** in history indicator area ***510a***. For query window **511**, there are two nodes at that level of the search path, indicated by boxes ***511b–511c*** of history area ***511a***. In this case, the box ***511c*** is in the search path, so it is highlighted. Finally, the box ***512b*** of the history area ***512a*** is highlighted since it is in the search path.

FIG. ***5c*** illustrates windows **520–522** which correspond to the three levels of the tree structure. Query window **520**

corresponds to the search scope of nodes **323** and **324**, query window **521** corresponds to the search scope of node **321** and query window **522** corresponds to the search scope of node **320**. Here, two search scopes at the same level are concurrently displayed. This is indicated by the highlighting of the boxes **520***a* and **520***b*. The results display area of query window **520** is divided into areas **523***a* and **523***b* each corresponding to a search scope. Such a display may be used for executing a query against both search scopes simultaneously.

History Window

The present invention combines the graphic visual representation of the results of a query with a means for maintaining and traversing a search history. The search history is used in the event that subsequent queries do not provide the desired results and it is desirable to restart at a convenient starting point. The search history in the currently preferred embodiment is generated in a hierarchical tree structure wherein each node represents a search scope. In this structure, a child node will represent a subset of the scope of the parent node. A user determines when a new search scope and resulting node is created. The entire search history is presented in a history window.

The history window is independent of the query windows. A history window is illustrated in FIG. **6**. Referring to FIG. **6**, the tree structure of FIG. **3** is shown on history window **601**. The present invention provides for direct movement to the various query windows from the history window. This is done in a point and click fashion. By pointing to the node representing the desired query window and clicking on the cursor control button, the desired query window can be displayed to the user. Other functions such as deleting query windows can be performed form the history tree.

Techniques for graphical creation, representation and manipulation of tree structures are known in the art. Any such techniques could be implemented for use with the present invention.

Visual Representation of Query Results

The present invention provides a graphical display output which allows a user to visualize the results of a query to a database beyond a mere list format. Query results are graphically presented as selectable disjoint subsets. Two visualizations described below are the venn diagram and the perspective wall. A user may choose which visualization is used in connection with a particular query window, or the visualization may be selected automatically based on the query. Alternatively, a user may wish to have the same query results displayed using the various visualization techniques. Choosing which visualization to use may occur either before or after the query is executed. Each of these visualizations and a corresponding query window are now described.

Venn Diagrams

A Venn diagram uses circles to represent sets of data. Position and overlap of the circles indicate logical relationships between the sets of data. A Venn diagram based visualization is premised on the number of dimensions supported by the computer controlled display system. This is because no more than n+1 mutually intersecting sets may be readily displayed in n-space. So for example if the computer controlled display system generates graphical information in an n=2 space, (i.e. two dimensions) the number of sets or expressions that may be visualized is limited to three. Naturally, if n=3, up to 4 expressions or sets may be utilized. The currently preferred embodiment illustrates the case of n=2. However, this does not limit the number of expressions that can be used for a search since queries can be nested.

Query windows embodying a venn diagram implementation are illustrated in FIGS. **7–9**. The creation of the query windows illustrated in FIGS. **7–9** may be done utilizing programming tools or toolkits generally available to application program developers. FIG. **7** illustrates a query window prior to a query being made. Referring to FIG. **7**, a query window **701** includes a description of the search scope **702** and a plurality of query input areas **704–706**. It should be noted that each of the query input areas **704–706** has a different visually distinctive attribute. This visually distinctive attribute may be a color or a fill pattern. In FIG. **7**, the query input area **704** has a vertical lines fill pattern, the query input area **705** has a horizontal lines fill pattern and the query input area **706** has a right slanted lines fill pattern. Because each of these queries is visually distinctive, the results of the various expressions in a query may be readily determined. Finally, the query window **701** includes history indicator area **703***a*. The history indicator area **703***a* is used to indicate a location (i.e. level) in the query history. The visual appearance of box(es) (e.g. box **703***b*) displayed the history indicator area **703***a* will also be an indicator of whether the corresponding search scope is in the current path of a query.

When querying the database, a user will enter expressions of the query into query input areas **704–705**. The query would then be executed using a mechanism such as depressing the enter or return key of a keyboard coupled to the computer controlled display system, via a menu item, or via some switch or button (e.g. an execute button invoked by a point and click function) in the query window itself. FIG. **8** is a screen display illustrating the results of a query. Referring to FIG. **8**, first, second and third search expressions **801–803** have been entered into query input areas **704–706**, respectively. As a result of executing the query, a textual description of the results of the query for each of the expressions **804**, is provided. This may include the actual number of elements responsive to the search query.

Further included in the query window is a results display area displaying a Venn diagram. In FIG. **8**, the Venn diagram includes circles **805–807**, which correspond to the results of search expressions **801–803**, respectively. The number of documents satisfying the search expression of the corresponding circle is indicated by its size, by a number contained in the circle, or both. Each of the circles may also contain a list of or some iconic representation of the documents satisfying the corresponding search expression. It is significant to note that the resultant circles of the Venn diagram has the same fill pattern as found in the corresponding query input area. This allows a quick visual interpretation of how the query expressions relate.

The disjoint selectable subsets of the venn diagram visualization are further illustrated in FIG. **8**. Note that overlap area **808** indicates the intersection of circles **805** and **806**, the overlap area **809** indicates the intersection of circles **805** and **807**, the overlap area **810** indicates the intersection of circles **806** and **807**, and the overlap area **811** indicates the intersection of each of the circles **805–807**. Each of the areas of the circles that do not overlap another circle are exclusive from the other circles (i.e. there are no common results). Each of these areas, as well as an entire circle, would constitute a selectable disjoint subset.

FIG. **9** illustrates the results of a different query. Referring to FIG. **9**, a set of expressions **901–903** are entered into the

query input areas **704–706** respectively. The resulting circles **904–906** of the Venn diagram correspond to the expressions **901–903** From the resulting Venn diagram in FIG. **9**, it can be readily observed that no items in the database in circle **906** are in any of the other circles. This could be a desired or undesired result. In any event, it is clear that the relationships of the results between the different expressions can be quickly ascertained.

In FIG. **9** the disjoint subsets are the overlap area **907**, the portion of circle **904** that does not overlap with circle **905**, the entire circle **904**, the portion of circle **905** that does not overlap with circle **904**, the entire circle **905** and the entire circle **906**.

Selection of a new search scope is a straightforward task. A user would simply point and click to the disjoint subsets in the results display to include those items are included in the new search scope. Selection of each of the overlap areas would be for the items satisfying the logical relationship indicated by the overlaps. Selection techniques for getting either all of what is in a particular circle, or the part of a circle not overlapping with another circle would be relatively straightforward. For example, a double click in a non-overlapping portion of a circle could be used to indicate selection of the entire circle, whereas a single click could indicate only the portion of the circle that does not overlap with another circle.

Perspective Wall

The perspective wall visualization permits a user to lay out search results along a linear property, such as date or version. The perspective wall is described in EP 0447 095A, Robertson, et al., entitled "Workspace Display". In the perspective wall visualization, database search results are organized along two user defined axes. So for example, a user may select date as a horizontal axes and author as a vertical axes. Items associated with a particular author are then laid out along the wall in time order. A query window for the perspective wall visualization is illustrated in FIG. **10**. Referring to FIG. **10**, the results of a query are organized onto perspective wall **1001** in the result display area **1002** of query window **1003**. The results of a query are mapped to the wall according to user provided criteria (defined below). Traversal along the wall is invoked by scrolling to different parts of the wall or by "stretching" and "destreching" which is described in EP 0447 095A. The input area **1004** is different in that the intent of the perspective wall visualization is to filter out portions of the search scope and then organize along a linear property. So a first input box **1005**, labeled linear property, is for identifying a property of the search scope to which the subsequent search results are laid out. This can be though of as the horizontal axis property. A second input box **1006**, labeled grouping property, is for identifying a property of the search results by which the data linearly grouped. This can be through of a vertical axis property. A third input box **1007** is for providing one or more search expressions.

FIG. **11** is an example of a perspective wall visualization. Referring to FIG. **11**, the wall is comprised of a plurality of results laid out on the wall in a time and author fashion. Note that in the first input box **1005**, the property "date" has been entered and in second input box **1006**, the grouping property "author" has been entered. An expression "expression PW" has been entered into the third input box. On the perspective wall visualization, the documents satisfying the query criteria for authors "Hoppe", "Rao" and "Mackinlay" are laid out accordingly in time fashion. For the author "Hoppe" there are three documents dated Jan. 1 (as indicated at **1101**) and two documents dated Jan. 30 (as indicated at **1102**). The visualization indicates that there are other documents associated with the author "Hoppe", but the perspective wall must be traversed to get detail on those documents.

For the author "Rao", a single document is dated Jan. 1 (as indicated at **1103**), two documents dated Jan. 15 (as indicated at **1104**) and three documents dated Feb. 15 (as indicated at **1105**). The visualization indicates that there are other documents associated with the author "Rao", but the perspective wall must be traversed to get detail on those documents.

For the author "Mackinlay", a single document is dated Jan. 15 (as indicated by **1106**), two documents are dated Jan. 30 (as indicated by **1107**), two documents are dated Feb. 1 (as indicated by **1108**) and a single document is dated Feb. 15 (as indicated by **1109**). The visualization indicates that there are other documents associated with the author "Mackinlay", but the perspective wall must be traversed to get detail on those documents.

For the perspective wall visualization, the disjoint subsets would be the space between two points along the defined linear property, or the collection of documents satisfying one of the grouping properties or a collection of documents satisfying one of the grouping properties and is between two points along the defined linear property. Selection of a disjoint subsets to create a new search scope can be accomplished by a point and click operation at a start point on the wall and a point and click operation at and end point on the wall (e.g. between two dates). Of course, the user may traverse the wall to get to the desired start and end points. All the search results between the two dates would then be part of a new search scope. Selection may also be performed on a group. A group may be selected by a point and click operation on a label identifying that group. Finally, selection can be performed on a group(s) but for only those documents between two points on the wall. This may be accomplished by "double-clicking" on the desired group and then single clicking for the start and end points on the wall.

Thus, a computer controlled display system providing for graphical representation of a query to a database and creation and traversal through a search history is disclosed. While the embodiments disclosed herein are preferred, it will be appreciate from this teaching that various alternative, modifications, variations or improvements therein may be made by those skilled in the art, which are intended to be encompassed by the following claims.

What is claimed:

1. A computer controlled display system for displaying the results of a search for documents stored in a database, said computer controlled display system comprising:

a display for displaying a plurality of query windows;

means for defining a search scope;

means for generating a query window responsive to definition of a search scope, said query window comprising an input area for input of query expressions, a query results area for graphical display of query results, a history indicator area for displaying one or more search scope indicators, and a search scope area for indicating the search scope for the query window, said means for generating a query window coupled to said display;

means for entering a query to said database, said query comprised of one or more expressions;

query processing means for processing query expressions from said input areas of said query window and causing

display of the results of said query according to a user selected information visualization technique, said user selected information visualization technique causing display of a set of query results as a plurality of selectable disjoint subsets in said query results area, said query processing means coupled to said means for generating a query window.

2. The computer controlled display system as recited in claim 1 wherein said one or more search scope indicators of said history indicator area of said query window represents a search scope at a level in said search path.

3. The computer controlled display system as recited in claim 2 wherein one of said one or more search scope indicators of said history indicator area is highlighted to indicate the search scope being displayed.

4. The computer controlled display system as recited in claim 2 wherein two or more of said search scope indicators of said history indicator area are highlighted to indicate search scopes being displayed.

5. The computer controlled display system as recited in claim 2 wherein each of said one or more search scope indicators of said history indicator area is an icon.

6. The computer controlled display system as recited in claim 2 wherein each of said one or more search scope indicators of said history indicator area is one or more text symbols.

7. The computer controlled display system as recited in claim 1 wherein said information visualization technique is a venn diagram wherein each circle corresponds to one of said one or more expressions of said query and the spatial locations and overlaps of said circles define a plurality of selectable areas.

8. The computer controlled display system as recited in claim 7 wherein said means for defining a search scope is comprised of means for selecting one or more areas in said venn diagram.

9. The computer controlled display system as recited in claim 1 wherein said information visualization technique is a perspective wall.

10. The computer controlled display system as recited in claim 9 wherein said means for defining a search scope is comprised of means for selecting a first point on said wall and a second point on said wall.

11. The computer controlled display system as recited in claim 1 further comprising means for displaying a search path, said search path comprised of a plurality of query windows which are concentrically displayed so that the history indicator area for each query window is displayed.

12. The computer controlled display system as recited in claim 1 wherein said displays means is further for displaying a history window, said history window for displaying a history structure of a search history of said search for documents stored in said database.

13. The computer controlled display system as recited in claim 12 further comprising means for adding a node to said history structure in response to definition of a search scope.

14. The computer controlled display system as recited in claim 1 wherein said means for defining a search scope is comprised of means for selecting one or more of said plurality of selectable disjoint subsets in said query result area.

15. On a computer system having a display means and coupled to a database, a method for displaying the results of a query to said database on said display means, said method comprising the steps of:

a) displaying a first query window having a first search scope, said first query window having a query entry area, a results display area and a history indicator area;

b) executing a provided query from a user, said query comprised of a plurality of expressions entered in to said query entry area of said first query window;

c) displaying an information visualization of the set of results of said query in said results display area of said first query window, said information visualization displaying a plurality of selectable disjoint subsets;

d) determining that said user has selected one or more of said selectable disjoint subsets;

e) creating a second query window having a second search scope comprised of said selected one or more disjoint subsets, said query window having a query entry area, a results display area and a history indicator area; and

f) displaying said second query window concentric with said first query window so that the history indicator area of said first query window and the history indicator area of said second query window are concurrently displayed.

16. The method as recited in claim 15 wherein concurrent with step a), performing the step of creating a first node in a hierarchical history structure.

17. The method as recited in claim 16 wherein concurrent with step e), performing the step of creating a second node in a hierarchical history structure.

18. The method as recited in claim 17 wherein said method is further comprised of the steps:

g) detecting that said user has requested viewing of a history window, said history window for displaying said hierarchical history structure; and

h) displaying said history structure in a history window.

* * * * *

REF in SERIAL NO. 369,360

US005895470A

United States Patent [19]

Pirolli et al.

[11] Patent Number: 5,895,470

[45] Date of Patent: Apr. 20, 1999

[54] **SYSTEM FOR CATEGORIZING DOCUMENTS IN A LINKED COLLECTION OF DOCUMENTS**

[75] Inventors: **Peter L. Pirolli**, El Cerrito; **James E. Pitkow**, Palo Alto; **Ramana B. Rao**, San Francisco, all of Calif.

[73] Assignee: **Xerox Corporation**, Stamford, Conn.

[21] Appl. No.: 08/842,926

[22] Filed: Apr. 9, 1997

[51] Int. Cl.[6] G06F 17/30

[52] U.S. Cl. 707/102; 707/4; 707/101; 707/103; 707/5; 707/6; 707/104

[58] Field of Search 707/4, 101, 103, 707/5, 6, 102, 104

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,193,185	3/1993	Lanter	707/101
5,418,948	5/1995	Turtle	707/4
5,442,778	8/1995	Pedersen et al.	707/5
5,687,364	11/1997	Saund et al.	707/5
5,754,939	5/1998	Herz et al.	455/4.2
5,758,347	5/1998	Lo et al.	707/103
5,778,368	7/1998	Hogan et al.	707/10

FOREIGN PATENT DOCUMENTS

0632367 A1	1/1995	European Pat. Off.	3/6
2318479	4/1998	United Kingdom	12/23

OTHER PUBLICATIONS

Classification and Indexing Languages in Poland (1974–1986). Int. Classification. V 14. No. 1., pp. 23–28. Bielicka et al., Int. Jul. 1987.

The Use of Cluster Hierarchies in Hypertext Information Retrieval. Crouch et al., Hypertext '89 Proceedings, pp. 225–237. Nov. 1989.

Do We Need a Common Standard for the Design Structure Model?. Siepmann. Electronic Design Automation Frameworks. Elsevier Science Publishers—North Holland. IFIP 1991. pp. 349–364. Jan. 1991.

Self-documenting systems: a role for machine-aided indexing. Griffiths. The University of Tennesse. USA. Online Information 92. Dec. 8–10. 1992 London. England. pp. 291–296. Aug. 1992.

IN Search of Meta-knowledge. Lopez. Jr., Mathematical Sciences. Loyola University. email: lopez@loynovm.bitnet. pp. 263–269. N93–25983. May 1993.

Creating a Web Analysis and Visualization Environment. Kent et al., Consumer Networks and ISDN Systems. pp. 109–117 vol. 28. Jul. 1995.

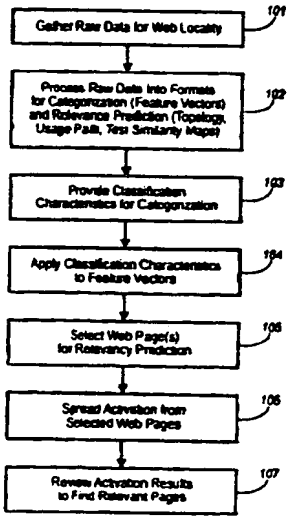
Search and Ranking Algorithms for Locating Resources on the World Wide Web. Yuwono et al., 1996 IEEE. pp. 164–171. Feb. 1996.

Primary Examiner—Wayne Amsbury
Assistant Examiner—Shahid Alam
Attorney, Agent, or Firm—Richard B. Domingo

[57] **ABSTRACT**

A system for extracting and analyzing information from a collection of linked documents at a locality to enable categorization of documents and prediction of documents relevant to a focus document. The system obtains and analyzes topology, usage and path information from for a collection at a locality, e.g. a web locality on the world wide web. For categorization, document meta information is represented as document vectors. Predefined criteria is applied to the document vectors to create lists of "similar" types of documents. For relevance prediction, networks representing topology, usage path and text similarity amongst the documents in the collection are created. A spreading activation technique is applied to the networks starting at a focus document to predict the documents relevant to the focus document. Using category and relevance prediction information, tools can be built to enable a user to more efficiently traverse through the collection of linked documents.

14 Claims, 10 Drawing Sheets



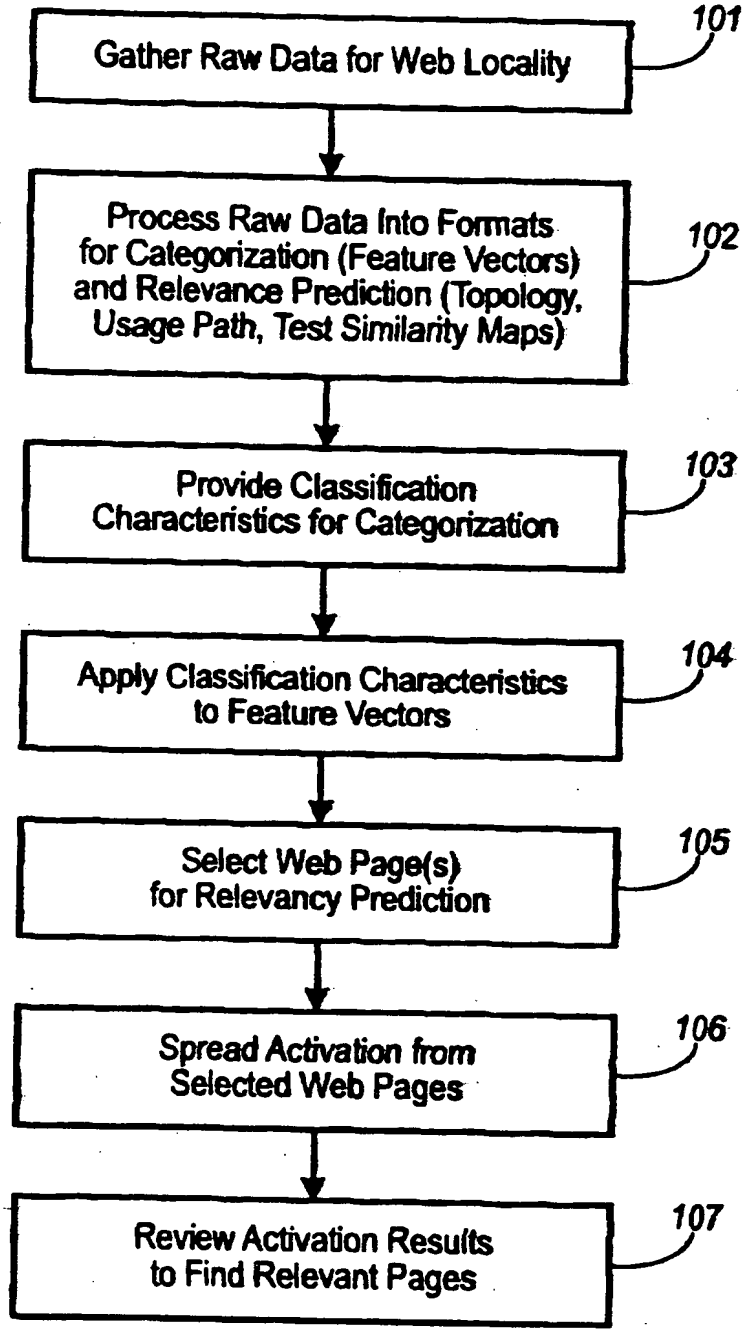

Gather Raw Data for Web Locality
101
Process Raw Data Into Formats for Categorization (Feature Vectors) and Relevance Prediction (Topology, Usage Path, Test Similarity Maps)
102
Provide Classification Characteristics for Categorization
103
Apply Classification Characteristics to Feature Vectors
104
Select Web Page(s) for Relevancy Prediction
105
Spread Activation from Selected Web Pages
106
Review Activation Results to Find Relevant Pages
107

FIG. 1

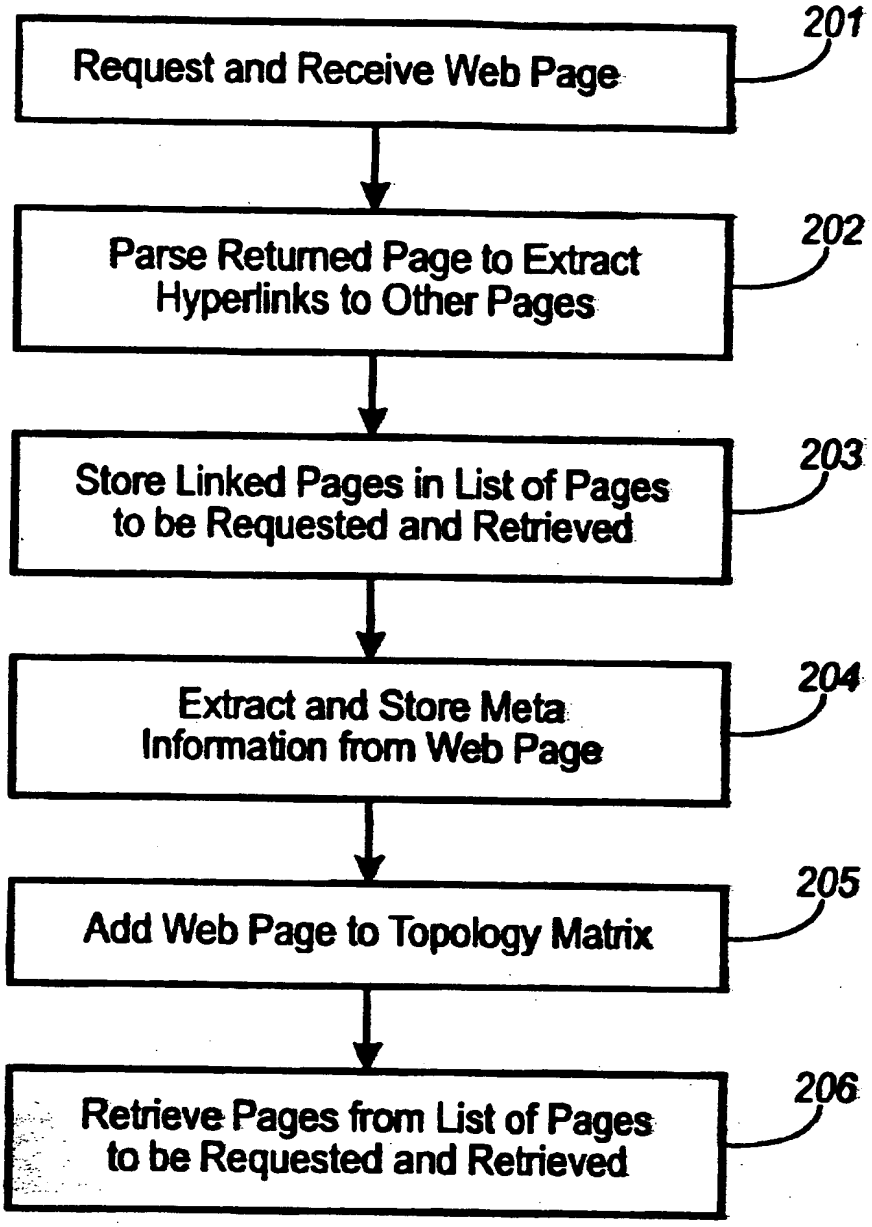

Request and Receive Web Page
201
Parse Returned Page to Extract Hyperlinks to Other Pages
202
Store Linked Pages in List of Pages to be Requested and Retrieved
203
Extract and Store Meta Information from Web Page
204
Add Web Page to Topology Matrix
205
Retrieve Pages from List of Pages to be Requested and Retrieved
206

FIG. 2

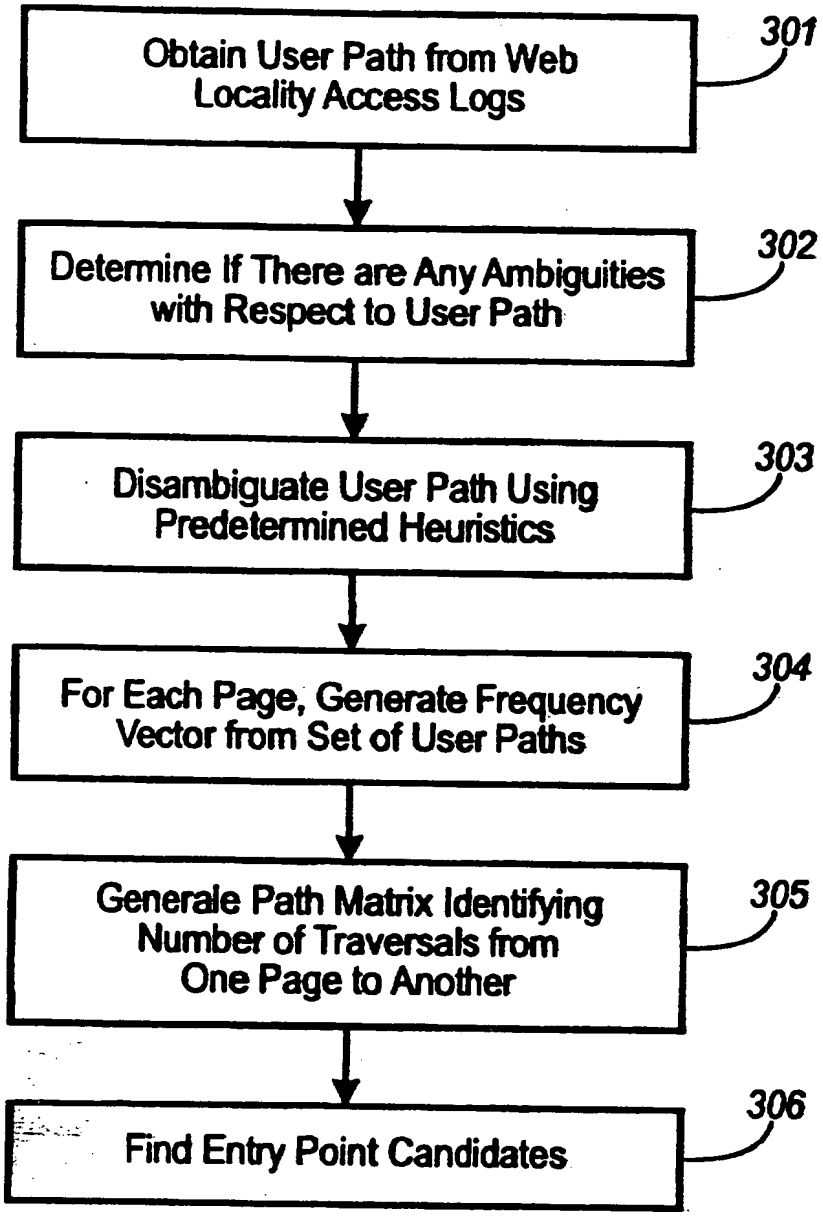

Obtain User Path from Web Locality Access Logs
301
Determine If There are Any Ambiguities with Respect to User Path
302
Disambiguate User Path Using Predetermined Heuristics
303
For Each Page, Generate Frequency Vector from Set of User Paths
304
Generale Path Matrix Identifying Number of Traversals from One Page to Another
305
Find Entry Point Candidates
306

FIG. 3

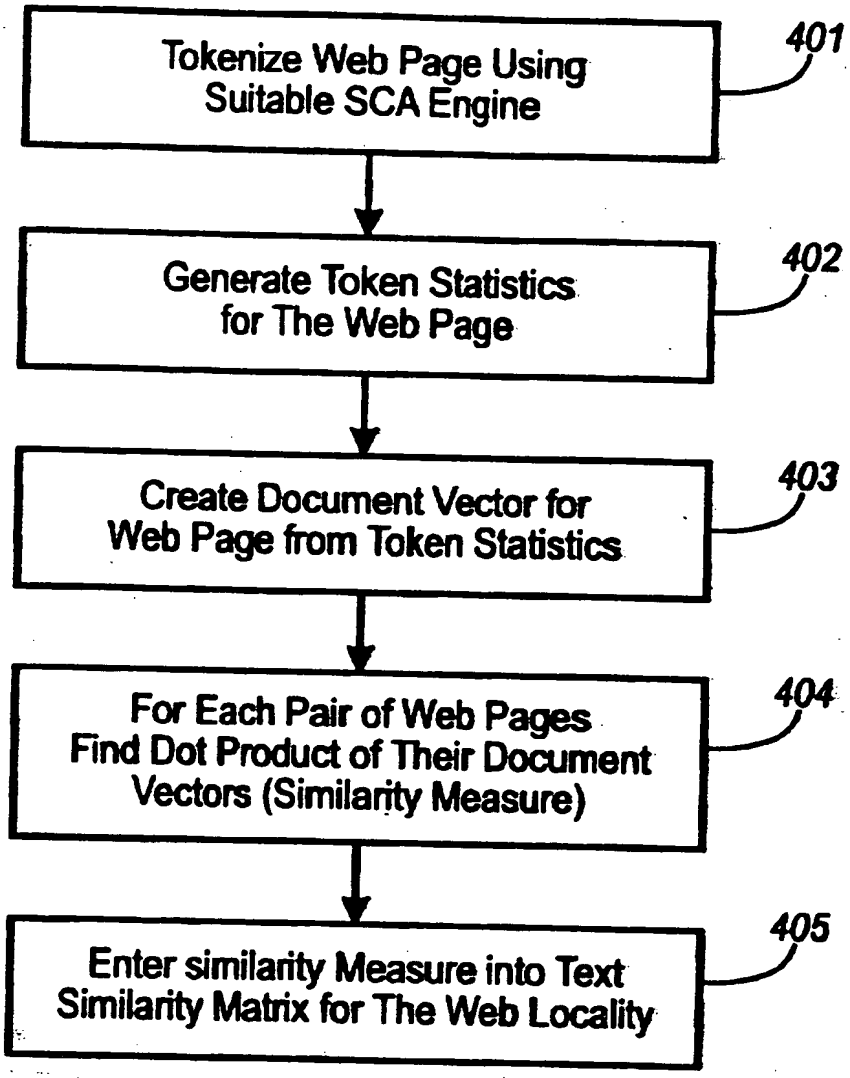

Tokenize Web Page Using Suitable SCA Engine
401
Generate Token Statistics for The Web Page
402
Create Document Vector for Web Page from Token Statistics
403
For Each Pair of Web Pages Find Dot Product of Their Document Vectors (Similarity Measure)
404
Enter similarity Measure into Text Similarity Matrix for The Web Locality
405

FIG. 4

500

	502 Page Identifier	503 Size (Bytes)	504 Inlinks	505 Outlinks	506 Frequency	507 Sources	508 CSIM	509 CDEPTH
501	Page 1	500	8	7	113	105	95	5
	Page 2	1500	2	3	45	0	56	31
	Page 3	460	-5	5	16	3	38	2
	Page 4	1200	4	3	78	56	77	2
	Page 5	479	1	2	23	20	69	3
	Page 6	2267	6	5	100	77	98	2
	Page 7	3397	3	2	90	2	76	1
	Page 8	6501	1	1	88	0	0	0

FIG. 5

	Page Category	Size	Number Inlinks	Number Outlinks	Depth of Children	Similarity to Children	Frequency	Entry Point
604	Index	(outlinks /size)	0	+1	0	0	0	0
605	Source Index	(outlinks /size)	0	+1	0	0	0	+1
606	Reference	+1	-1	-1	-1	0	0	0
607	Destination Reference	+1	-1	-1	-1	0	0	-1
601	Head	0	0	+1	+1	+1	0	+1
602	Organization Home Page	0	+1	+1	0	+1	0	+1
603	Personal Home Page	>1000k <3000k	0	0	0	0	-1	-1
608	Content	+1	-1	-1	0	0	0	0

FIG. 6

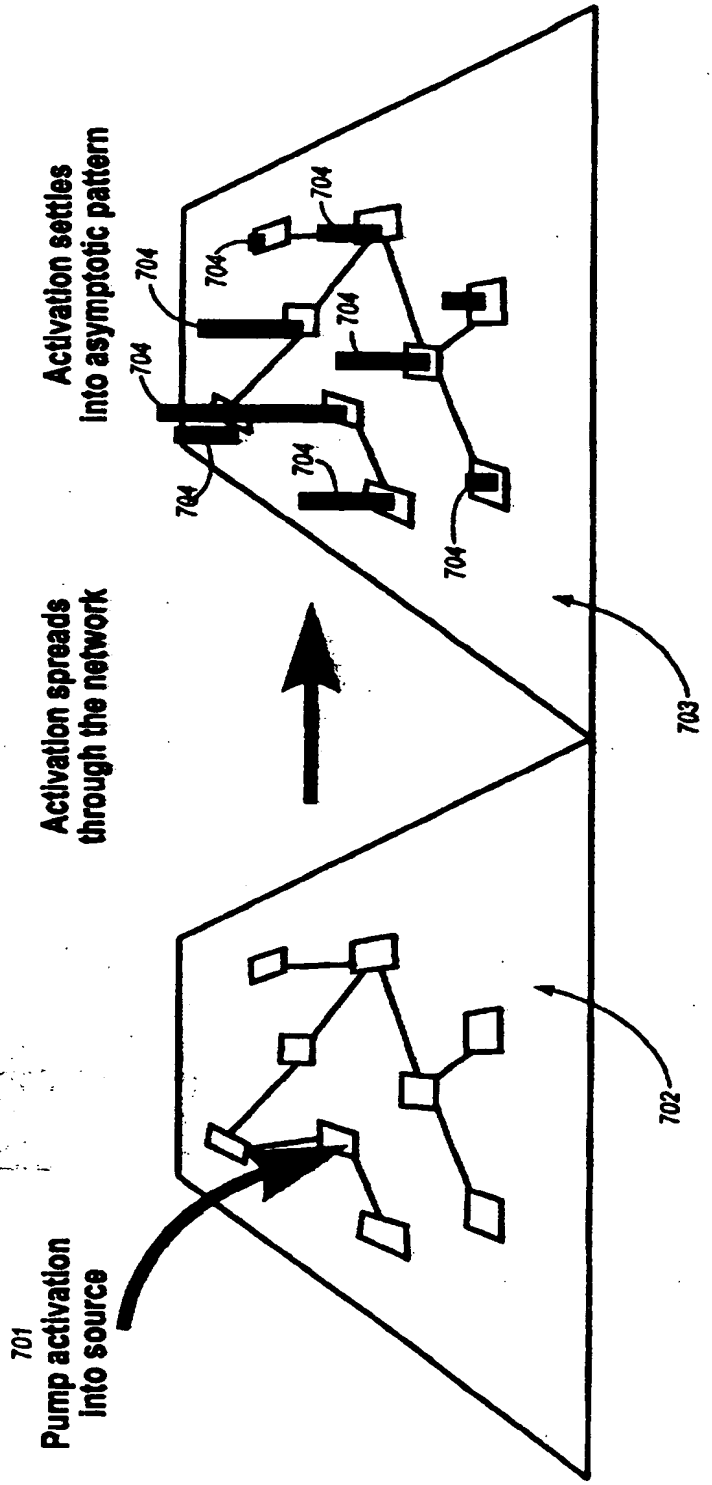

Activation settles into asymptotic pattern
Activation spreads through the network
Pump activation into source
701
702
703
704

FIG. 7

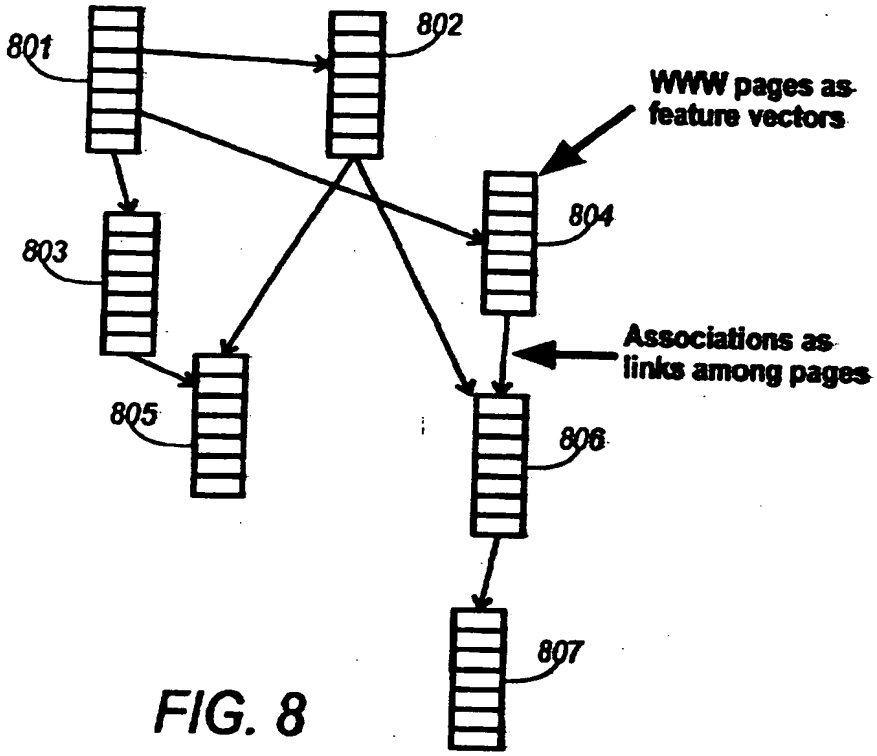


FIG. 8

Pages

Pages	801	802	803	804	805	806	807
801	0	1	1	1	0	0	0
802	0	0	0	0	1	1	0
803	0	0	0	0	1	0	0
804	0	0	0	0	0	1	0
805	0	0	0	0	0	0	0
806	0	0	0	0	0	0	1
807	0	0	0	0	0	0	0

FIG. 9

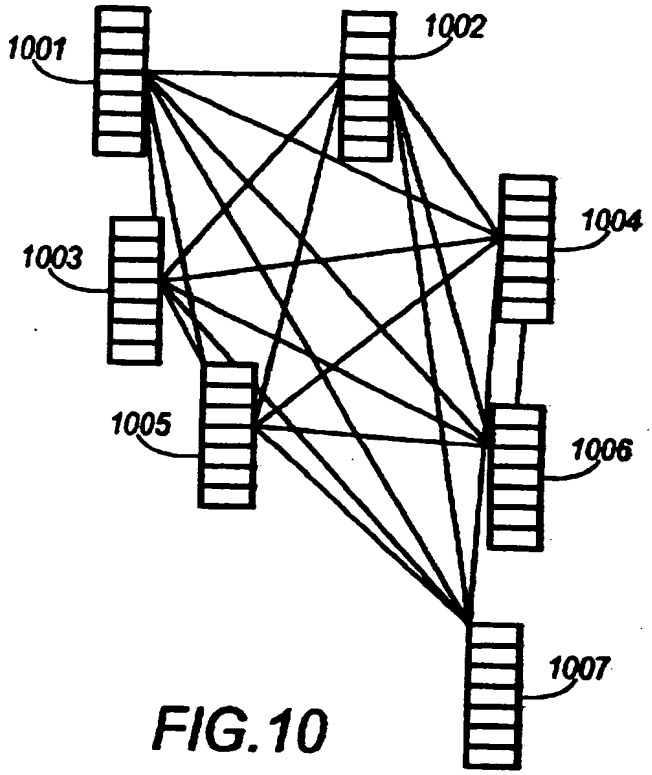


FIG.10

Pages	1001	1002	1003	1004	1005	1006	1007
1001	1	.5	.8	.1	.3	.7	.4
1002	.2	1	.1	.7	.6		
1003			1				
1004				1			
1005					1		
1006						1	
1007							1

FIG.11

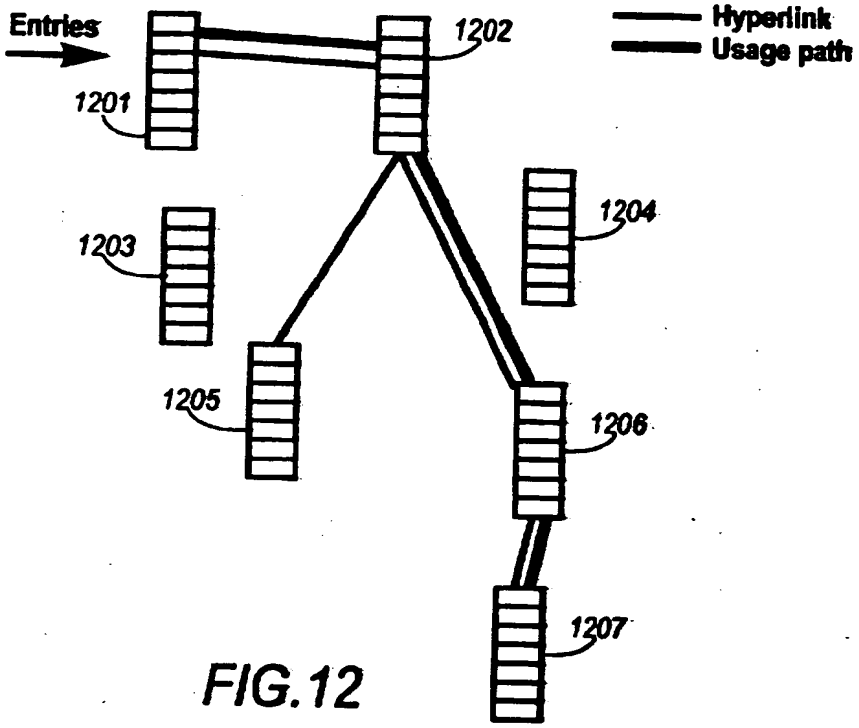


FIG.12

Pages

Pages	1201	1202	1203	1204	1205	1206	1207
1201	0	20	0	0	0	0	0
1202	0	0	0	0	10	15	0
1203	0	0	0	0	0	0	0
1204	0	0	0	0	0	0	0
1205	0	0	0	0	0	0	0
1206	0	0	0	0	0	0	7
1207	0	0	0	0	0	0	0

FIG.13

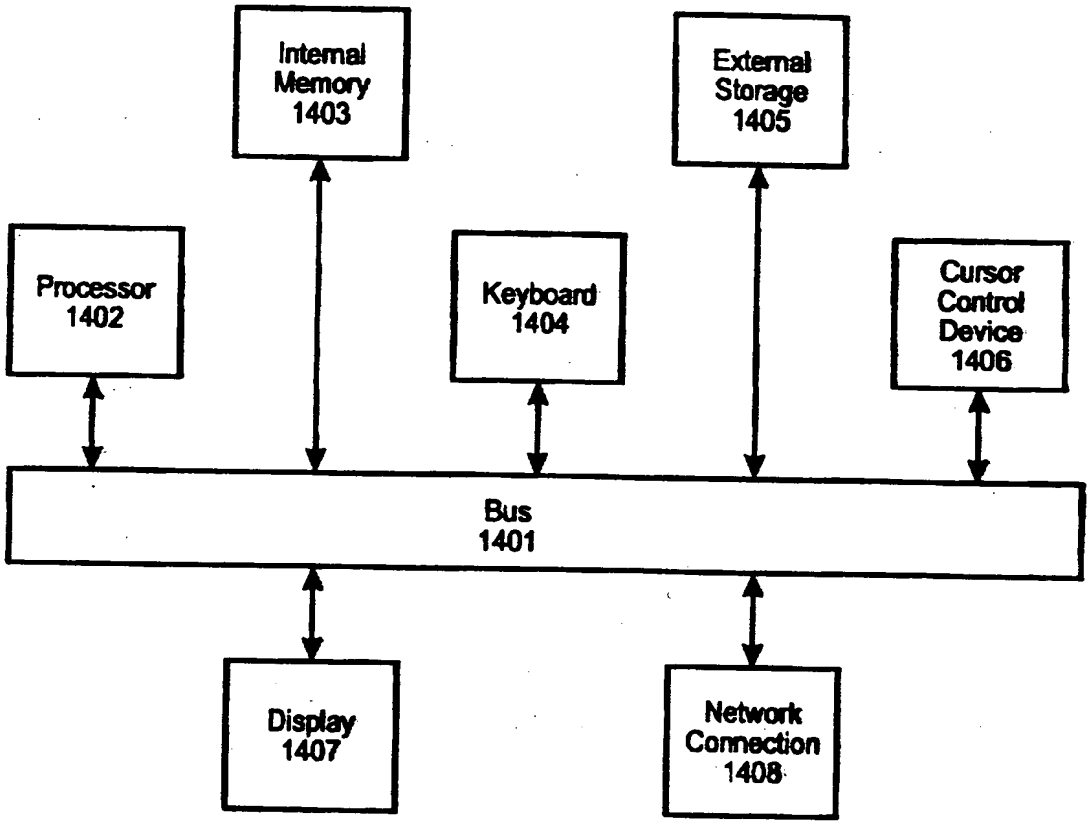

Internal Memory 1403
External Storage 1405
Processor 1402
Keyboard 1404
Cursor Control Device 1406
Bus 1401
Display 1407
Network Connection 1408

FIG.14

SYSTEM FOR CATEGORIZING DOCUMENTS IN A LINKED COLLECTION OF DOCUMENTS

CROSS REFERENCE TO RELATED APPLICATIONS

The present application is related to commonly assigned U.S. patent application Ser. No. 08/836,807 entitled "System For Predicting Documents Relevant To Focus Documents By Spreading Activation Through Network Representations Of A Linked Collection Of Documents" U.S. Pat. No. 5,835,905 which was filed concurrently with the present application.

FIELD OF THE INVENTION

The present invention is related to the field of analysis and design of linked collections of documents, and in particular to categorization of documents in said collection.

BACKGROUND OF THE INVENTION

Users of large linked collections of documents, for instance as manifest on the World Wide Web, are motivated to improve the rate at which they gain information needed to accomplish their goals. Hypertext structures primarily affords information seeking by the sluggish process of browsing from one document to another along hypertext links. This sluggishness can be at least partly attributed to three sources of inefficiency in the basic process. First, basic hypertext browsing entails slow sequential search by a user through a document collection. Second, important information about the kinds of documents and content contained in the total collection cannot be immediately and simultaneously obtained by the user in order to assess the global nature of the collection or to aid in decisions about what documents to pursue. Third, the order of encounter with documents in basic browsing is not optimized to satisfy users' information needs. In addition to exacerbating difficulties in simple information-seeking, these problems may also be found in the production and maintenance of large hypertext collections.

There are two widely visible technologies that may be considered broadly as seeking to address the above inefficiencies:

- Text-based information retrieval techniques that rapidly evaluate the predicted relevance of documents to a user's topical query (e.g. services such as Alta Vista™, LycoS™, and Infoseek® which operate on the World Wide Web). This effectively changes slow sequential search to nearly parallel search, and provides an improved ordering of the users' search through documents.
- Community/service categorization of documents. For instance, this service is provided by Yahoo™, which has a hierarchy of Web pages that define a topic taxonomy.

Known previous work has focused on attempts to extract higher level abstractions which can be used to improve navigation and assimilation of hypertext. Such work has typically used topological or textual relationships to drive analysis.

SUMMARY OF THE INVENTION

A system for analyzing the topology, content and usage of linked collections of documents such as those found on the World Wide Web (hereinafter the Web) to facilitate information searching or improving design of a web locality is disclosed. Documents found on the Web are typically referred to as Web pages. The system provides for (a) categorization based on feature vectors that characterize individual page information and (b) prediction of need (or relevance) of other Web pages with respect to a particular context, which could be a particular page or set of pages, using a spreading activation technique. In combination, these provide (from the user's perspective) nearly-parallel search, simultaneous identification of the types of all documents in a collection, and prediction of expected need. These techniques may be used in support of various information visualization techniques, such as the WebBook described in co-pending and commonly assigned application Ser. No. 08/525,936 entitled "Display System For Displaying Lists of Linked Documents", to form and present larger aggregates of related Web pages. Categorization techniques are based on representations of Web pages as feature vectors containing information about document content, usage, and topology, as well as content, usage, and topology relations to other documents. These feature vectors are used to identify and rank particular kinds of Web pages, such as "organization home pages" or "index pages."

Spreading activation techniques are based on representations of Web pages as nodes in graph networks representing usage, content, and hypertext relations among Web pages. Conceptually, activation is pumped into one or more of the graph networks at nodes representing some starting set of Web pages (i.e. focal points) and it flows through the arcs of the graph structure, with the amount of flow modulated by the arc strengths (which might also be thought of as arc flow capacities). The asymptotic pattern of activation over nodes will define the degree of predicted relevance of Web pages to the starting set of Web pages. By selecting the topmost active nodes or those above some set criterion value, Web pages may be aggregated and/or ranked based on their predicted relevance.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a flowchart illustrating the basic steps for web page categorization and relevance prediction as may be performed in the currently preferred embodiment of the present invention.

FIG. 2 is a flowchart illustrating the steps for obtaining the topology and meta-information for a web locality as may be performed in the currently preferred embodiment of the present invention.

FIG. 3 is a flowchart illustrating the steps for obtaining usage statistics, usage path and entry point information as may be performed in the currently preferred embodiment of the present invention.

FIG. 4 is a flowchart for calculating a text similarity matrix as may be performed in the currently preferred embodiment of the present invention.

FIG. 5 is an illustration of a feature vector as may be utilized in the currently preferred embodiment of the present invention.

FIG. 6 is a table showing examples of categories and the corresponding feature weightings for the categories as may be utilized in the currently preferred embodiment of the present invention.

FIG. 7 is a diagram illustrating the concept of spreading activation, as may be utilized in the currently preferred embodiment of the present invention.

FIG. 8 is an illustration of a topology network for a Web locality.

FEATURE VECTORS THAT CHAR INDIV. P. INFORMATION

PREDICTION OF NEED/ RELEVANCE OF OTHER WEB PAGES WRT... A PARTICULAR PAGE

FIG. 9 is an illustration of a matrix representation of the topology network of FIG. 8.

FIG. 10 is an illustration of a text similarity network for a Web locality.

FIG. 11 is an illustration of a matrix representation of the text similarity network of FIG. 10.

FIG. 12 is an illustration of a usage path network for a Web locality.

FIG. 13 is an illustration of a matrix representation of the usage path network of FIG. 12.

FIG. 14 is a block diagram illustrating the basic components of a computer based system as may be used to implement the currently preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

A system for analyzing the topology, content and usage of collections of linked documents is disclosed. The information derived from such a system may be used to aid a user in browsing the collection, redesigning the organization of the collection or in creating visualizations of the collections. The system provides a means for automatically categorizing the pages in the collection and a means for predicting the relevance of other pages in a collection with respect to a particular Web page using a spreading activation technique.

The currently preferred embodiment of the present invention is implemented for analyzing collections of linked documents residing on the portion of the Internet known as the World Wide Web (hereinafter the Web). However, it should be noted that the present invention is not limited to use on the Web and may be utilized in any system which provides access to linked entities, including documents, images, videos, audio, etc. The following terms defined herein are familiar to users of the Web and take on these familiar meanings:

World-Wide Web or Web: The portion of the Internet that is used to store and access linked documents.

Web Page or Page: A document accessible on the Web. A Page may have multi-media content as well as relative and absolute links to other pages.

Web Locality: A collection of related web pages associated with an entity having a site on the World-Wide Web such as a company, educational institute or the like.

Topology: The logical organization of web pages at a web locality as defined by links contained in the individual web pages.

Home Page: A page functioning as an entry point to a set of related pages on the Web. A home page will typically have a plurality of relative links to related pages.

Uniform Resource Locator (URL): The address or identifier for a page on the Web.

Server: An addressable storage device residing on the Internet which stores Web Pages.

Link: An indicator on a Web page which refers to another Web page and which can typically be retrieved in a point and click fashion. The Link will specify the URL of the other Web page.

Web Browser or Browser: A tool which enables a user to traverse through and view documents residing on the Web. Other rendering means associated with the Browser will permit listening to audio portions of a document or viewing video or image portions of a document.

Meta-information: Characteristic information for a particular Web page, including name, file size, number of links to pages in the Web locality, number of links to pages outside of the Web locality, depth of children, similarity to children, etc.

Overview

To best understand the context of the present invention, assume a scenario in which a user searches for relevant, valuable information at some web locality. The optimal selection of Web pages from the web locality to satisfy a user's information needs depends, in part, on the user's ability to rapidly categorize the Web page types, assess their prevalence on the web locality, assess their profitabilities (amount of value over cost of pursuit), and decide which categories to pursue and which to ignore. The overall rate of gaining useful information will be improved by eliminating irrelevant or low-value categories of information from consideration. Simply put, a user's precious time and attention benefits by being able to rapidly distinguish junk categories from important ones. This is improved by the degree to which Web pages can be quickly and simultaneously categorized.

Memory systems, whether human or machine, serve the purpose of providing useful information when it is needed. In part, the design of such systems is adaptive to the extent that they can reduce the costs of retrieving the information that is likely to be needed in a given context. This, for instance, is what memory caches and virtual memory attempt to optimize. For contexts involving human cognition, it has been argued that three general sorts of information determine the need probabilities of information in memory, given a current focus of attention: (1) past usage patterns, (2) degree of content shared with the focus, and (3) inter-memory associative link structures. The Web can be viewed as an external memory and a user would be aided by retrieval mechanisms that predicted and returned the most likely needed Web pages, given that the user has indicated an interest in a particular Web page in the Web locality.

In the present invention a kind of spreading activation mechanism is used to predict the needed Web page(s), computed using past usage patterns, degree of shared content, and the Web topology. The present invention utilizes techniques for inducing such information, and for approximating the computation of need probabilities using spreading activation. Also described is a way of precomputing a base set of spreading activation patterns from which all possible patterns can be computed in a simple and efficient way (whose cost is proportional only to the number of activation sources involved in a retrieval).

The basic steps for categorizing web pages in a web locality and for predicting relevance of other pages of a selected page as may be performed in the currently preferred embodiment of the present invention are briefly described with reference to the flowchart in FIG. 1. First, raw data is gathered for the web locality, step 101. Such raw data may be obtained from usage records or access logs of the web locality and by direct traversal of the Web pages in the Web locality. As described below, "Agents" are used to collect such raw data. However, it should be noted that the described agents are not the only possible method for obtaining the raw data for the basic feature vectors. It is anticipated that Internet service providers have the capabilities to provide such raw data and may do so in the future.

In any event, the raw data is then processed into desired formats for performing the categorization (feature vectors) and relevance prediction (topology, usage path and text similarity maps), step 102. The raw data is comprised of topology information, page meta-information, page frequency path information and text similarity information. Topology information describes the hyperlink structure among Web pages at a Web locality. Page meta-information

defines various features of the pages, such as file size and URL. Usage frequency and path information indicate how many times a Web page has been accessed and how many times a traversal was made from one Web page to another. Text similarity information provides an indication of the similarity of text among all text Web pages at a Web locality.

For the classification of Web pages in the web locality, classification characteristics are provided, step **103**. The classification characteristics are predetermined "rules" which are applied to the feature vectors of a page to determine the category of the page. For example, it may be desirable to have a classification of web pages as index types (contain primarily links to other pages) or content types (contain primarily information). The classification characteristics are then applied to the feature vectors representing the Web pages, step **104**. When the classification characteristics are applied to the respective feature vectors, lists of pages in the particular classes are created.

As noted above with respect to step **102**, topology, usage path and text similarity maps of the web locality are generated from the raw data. These maps represent the strength of association among web pages in the locality. The topology map indicates the hyper link structure of the web locality and are used to perform the relevance prediction. The usage path map indicates the flow or paths taken during traversal of the web locality. The text similarity map indicates similarity of content between pages in the web locality. These maps are used perform the relevancy predictions.

For relevancy predictions, one or more Web pages for spreading activation are selected, step **105**. The selected Web pages may be based on the category that it is in. Alternatively, if a user is currently browsing the pages in the web locality, the selected page may be the one currently being browsed. In any event, activation is spread using the selected page as a focal point to generate a list of relevant pages, step **106**. Generally, activation is pumped into one or more of the maps at the selected Web pages and it flows through the arcs of the maps, with the amount of flow modulated by the arc strengths (which might also be thought of as arc flow capacities). Review activation results to find relevant pages, step **107**. The asymptotic pattern of activation over nodes in the maps (i.e. Web pages) will define the degree of predicted relevance of Web pages to the selected set of Web pages. By selecting the topmost active Web pages or those above some set criterion value, Web pages may be ranked based on their predicted relevance. Subsequent traversal may then be performed based on the identified relevant Web pages.

Compiling the Raw Data for a Web Locality

Three basic kinds of raw data are extracted from a Web locality:

- Topology and meta-information, which are the hyperlink structure among Web pages at a Web locality and various features of the pages, such as file size and URL.
- Usage frequency and usage paths, which indicate how many times a Web page has been accessed and how many times a traversal was made from one Web page to another.
- Text similarity among all text Web pages at a Web locality

As described mentioned above with respect to FIG. 1, the raw data is used to construct two types of representations:

- Feature-vector representations of each Web page that represent the value of each page on each dimension and which are used in the categorization process
- Graph representations of the strength of association of Web pages to one another, which are used in the spreading activation. The graphs are represented using matrix formats.

Topology and Meta-information

The site's topology is ascertained via "the walker", an autonomous agent that, given a starting point, performs an exhaustive breadth-first traversal of pages within the Web locality. FIG. 2 is a flowchart illustrating the steps performed by the walker. Referring to FIG. 2, the walker uses the Hypertext Transfer Protocol (HTTP) to request and retrieve a web page, step **201**. The walker may also be able to access the pages from the local filesystem, bypassing the HTTP. The returned page is then parsed to extract hyperlinks to other pages, step **202**. Links that point to pages within the Web locality are added to a list of pages to request and retrieve, step **203**. The meta-information for the page is also extracted and stored, step **204**. The meta-information includes at least the following page meta-information: name, title, list of children (pages associated by hyperlinks), file size, and the time the page was last modified. The page is then added to a topology matrix, step **205**. The topology matrix represents the page to page hypertext relations, and a set of meta-information called the meta-document vectors, which represents the meta-information for each Web page. The list of pages to request and retrieve is then used to obtain the next page, step **206**. The process then repeats per step **202** until all of the pages on the list have been retrieved.

Thus, the walker produces a graph representation of the hyperlink structure of the Web locality, with each node having at least the above described meta-information. It is salient to note that the walker may not have reached all nodes that are accessible via a particular server—only those nodes that were reachable from the starting point (e.g. a Home Page for the Web locality) are included. This can be alleviated by walking the local filesystem the locality resides on.

Usage Statistics, Usage Paths, and Entry Points

Most servers have the ability to record transactional information, i.e. access logs, about requested items. This information usually consists of at least the time and the name of the URL being requested as well as the machine name making the request. The latter field may represent only one user making requests from their local machine or it could represent a number of users whose requests are being issued through one machine, as is the case with firewalls and proxies. This makes differentiating the paths traversed by individual users from these access logs non-trivial, since numerous requests from proxied and firewalled domains can occur simultaneously. That is, if 200 users from behind a proxy are simultaneously navigating the pages within a site, how does one determine which users took which paths? This problem is further complicated by local caches maintained by each browser and intentional reloading of pages by the user.

The technique implemented to determine user's paths, a.k.a. "the whitter", utilizes the Web locality's topology along with several heuristics. FIG. 3 is a flowchart illustrating the steps performed to determine user paths. First, a user path is obtained from the web locality access logs, step **301**. The topology matrix is consulted to determine legitimate traversals. It is then determined if there are any ambiguities with respect to the user path, step **302**. As described above such ambiguities may arise in the situation where the request is from a proxied or firewalled domain. If an ambiguity is suspected, predetermined heuristics are used to disambiguate user paths, step **303**. The heuristics used relies upon a least recently used bin packing strategy and session length time-outs as determined empirically from end-user naviga-

tion patterns. Essentially, new paths are created for a machine name when the time between the last request and the current request was greater than the session boundary limit, i.e., the session timed out. New paths are also created when the requested page is not connected to the last page in the currently maintained path. These tests are performed on all paths being maintained for that machine name, with the ordering of tests being the paths least recently extended. The foregoing analysis produces a set of paths requested by each machine and the times for each request.

From the set of paths, a vector that contains each page's frequency of requests is generated (i.e. a frequency vector), step **304**, along with a path matrix containing the number of traversals from one page to another, step **305**. In the currently preferred embodiment, the matrix is computed using software that identifies the frequency of all substring combinations for all paths.

Additionally, the difference between the total number of requests for a page and the sum of the paths to the page is computed to generate a set of entry point candidates, step **306**. The entry point candidates are the Web pages at a Web locality that seem to be the starting points for many users. Entry points are defined as the set of pages that are pointed to by sources outside the locality, e.g. an organization's home page, a popular news article, etc. Entry points might provide useful insight to Web designers based on actual use, which may differ from their intended use on a Web locality. Entry points also may be used in providing a set of nodes from which to spread activation.

Inter-document Text Similarity

Techniques from information retrieval can be applied to calculate a text similarity matrix which represents the inter-document text similarities among Web pages. In particular, for each Web page, the text is tokenized and indexed using a statistical content analysis process. An SCA engine processes text Web pages by treating their contents as a sequence of tokens and gathering collection and document level object and token statistics (most notably token occurrence). A contiguous character string representing a word is an example of a token. So in the currently preferred embodiment of the present invention, the Web pages in a Web locality are processed by the SCA engine to yield various indexes and index terms. A suitable process for analysis and tokenization of a collection of documents (or database) is described in section 5 of a publication entitled "An Object-Oriented Architecture for Text Retrieval", Doug Cutting, Jan Pedersen, and Per-Kristian Halvorsen, Xerox PARC technical report SSL-90-83.

FIG. **4** is a flowchart describing the steps for generating a text similarity matrix. Referring to FIG. **4**, a suitable SCA engine is used to tokenize a web page, step **401**. Token statistics for the web page are then generated, step **402**. These statistics include token occurrence. The token information is then used to create a document vector, where each component of the vector represents a word, step **403**. Entries in the vector for a document indicate the presence or frequency of a word in the document. The steps **401–403** are repeated for each Web page in the Web locality. For each pair of pages, the dot product of these vectors is computed, step **404**. The dot product which produces a similarity measure. The similarity measure is then entered into the appropriate location of the text similarity matrix for the Web locality, step **405**.

The currently preferred embodiment further provides a method for computing a "desirability" index for each Web page that "ages" over time. Using this, one can predict the number of hits a page will receive. What may also be provided is a "life-change" index, that also "ages" over time, that predicts the likelihood of Web pages being altered.

Categorization of Web Pages

In order to perform categorizations each Web page at the Web locality is represented by a vector of features constructed from the above topology, metainformation, usage statistics and paths, and text similarities. These Web page vectors are collected into a matrix. Such a matrix is illustrated in FIG. **5**. Referring to FIG. **5**, each row **501** of the matrix **500** represents a Web page. The columns in matrix **500** represent the page's:

- page identifier, identifies the particular web page (column **502**)
- size, in bytes, of the item (column **503**)
- inlinks, the number of hyperlinks that point to the item from the web locality (column **504**).
- outlinks, the number of hyperlinks the item contains that point to other items in the web locality (column **505**).
- frequency, the number of times the item was requested in the sample period (column **506**).
- sources, number of times the item was identified as the start of a path traversal (column **507**).
- csim, the textual similarity of the item to it's children based upon previous SCA calculation (column **508**).
- cdepth, the average depth of the item's children as measured by the number of '/' in the URL (column **509**). Note that the means and distributions of the feature values are normalized.

The present invention assumes that categories are designed by someone (application designer, webmaster, end user), in contrast to being automatically induced. These categories might be, for instance, socially defined genres (personal home page; product description), or personally defined categories of interest.

The present invention utilizes an approach based on weighted linear equations that define the rules for predicting degree of category membership for each page at a web locality. That is, equations are of the form

$$c_i = w_1 v_1 + w_2 v_2 + \ldots + w_n v_n$$

for all pages i in a Web locality, where the vj are the measured features of each Web page, and the wj are weights.

Example of Categories

Categorization techniques typically attempt to assign individual elements into categories based on the features they exhibit. Based on category membership, a user may quickly predict the functionality of an element. For instance, in the everyday world, identifying something as a "chair" enables the quick prediction that an object can be sat on. The techniques described herein will thus rely on the particular features that can be extracted about Web pages at a Web locality.

One may conceive of a Web locality as a complex abstract space in which are arranged Web pages of different functional categories or types. These functional categories might be defined by a user's specific set of interests, or the categories might be extracted from the collection itself through inductive technologies (e.g. Scatter/Gather techniques as described by Cutting, et al. in a publication entitled "Scatter/gather: A cluster-based approach to browsing large document collections". *Proceedings of SIGIR'92*, Jun. 1992.). An example category might be organizational home page. Typical members of the category would describe an organization and have links to many other Web pages, providing relevant information about the organization, its divisions or departments, summaries of its purpose, and so on.

In the currently preferred embodiment, a set of functional categories is defined. Each functional category was defined in a manner that has a graded membership, with some pages being more typical of a category than others, and Web pages may belong to many categories. FIG. 6 is a table illustrating the Web categories defined in the currently preferred embodiment of the present invention:

head **601**: Typically a related set of pages will have one page that would best serve as the first one to visit. Head pages have two subclasses:

organizational home page **602**: These are pages that represent the entry point for organizations and institutions, usually found as the default home page for servers, e.g., http://www.org/

personal home page **603**: Usually, individuals have only one page within an organization that they place personal information and other tidbits on.

index **604**: These are pages that server to navigate users to a number of other pages that may or may not be related. Typical pages in this category have the words "Index" or "Table of Contents" or "toc" as part of their URL.

source index **605**: These pages are also head nodes, those that are used as entry points and indices into a related information space.

reference **606**: A page that is used to repeatedly explain a concept or contains actual references. References also have a special subclasses:

destination reference **607**: In graph theory these are best thought of as "sinks", pages that do not point elsewhere but that a number of other pages point to. Examples include pages of expanded acronyms, copyright notices, and bibliographic references.

content **608**: These are pages whose purpose is not to facilitate navigation, but to deliver information.

FIG. 6 further shows the weights used to order Web pages for each of the categories outlined above. For example, it is hypothesized that Content Pages would have few inlinks and few outlinks, but have relatively larger file sizes. So the content classification criteria **608** used to determine this category of pages had a positive weight, +1, and negative weight, −1, on the inlink and outlink features. For Head Nodes (classification criteria **601**), being the first pages of a collection of documents with like content, it is expected that such pages will have high text similarity between itself and its children, and would have a high average depth of its children, and that it would be more likely to be an entry point based upon actual user navigation patterns.

It is noted that sometimes categories are formed which cannot be captured by such rules (i.e., the rules assume linearly separable categories and people sometimes form categories that are not linearly separable). However, the approach of the currently preferred embodiment has the advantage of being easy to compute and having simple combinatorics. This means that (a) the rules could be easily defined by the average end-user, (b) that membership in all core categories can be precomputed and stored as another feature on the feature vector (a computed feature as opposed to a basic feature) and (c) membership in a mixture of categories is just another weighted linear equation in which the features are categories.

Relevance Prediction Through Spreading Activation

With the above information, various predictions can be made as to pages relevant to a particular page. The "spreading activation" technique is used to make such a prediction.

Spreading activation can be characterized as a process that identifies knowledge predicted to be relevant to some focus of attention.

As noted above, the raw data provided by the web agents are massaged into three matrix structures representing the (a) link topology, (b) usage flow, and (c) interpage text similarity. The spreading activation technique used for relevance prediction assumes that one may identify a pattern of input activation that represents a pattern or focus of attention. For instance, the focus may be a specific Web page or a prototype of a category. Activation from this focus point(s) spreads through one or more of the three graphs and eventually settles into a stable pattern of activation across all nodes. The activation values are assumed to be the predicted relevance to the input focus (or the probability that a page will be needed given the pages in the input focus).

Spreading activation across the networks is described conceptually with reference to FIG. 7. Referring to FIG. 7, activation **701** is pumped into one or more of the graph networks **702** at nodes representing some starting set of focus Web pages. The activation flows through the arcs of the graph structure, with the amount of flow modulated by the arc strengths (which might also be thought of as arc flow capacities). The asymptotic pattern of activation over nodes, as illustrated by bars **704** contained in the nodes at activated network **703**, will define the degree of predicted relevance of Web pages to the starting set of focus Web pages. By selecting the topmost active nodes or those above some set criterion value, Web pages are extracted and ranked based on their predicted relevance.

The particular technique described has the property that the activation patterns that result from multiple input sources are just additive combinations of the activation patterns produced by each of the sources individually (multiple weighted sources are just weighted additions of the individual sources). Using this property, one may precompute the activation patterns that arise from each source combined with each graph. All complex patterns can be derived from these by simple vector addition. In addition, the activation values arising in each activation pattern can be combined with the categorization values.

In the currently preferred embodiment, the activation spreading technique used is a leaky capacitor model described by J. R. Anderson and P. L. Pirolli, in "Spread of Activation", *Journal of Experimental Psychology: Learning, Memory, and Cognition*, pp. 791–798 (1984) and by Huberman, B. A. and T. Hogg, in "Phase Transitions In Artificial Intelligence Systems", *Artificial Intelligences*, pp. 155–171(1987).

Networks for Spreading Activation

As outlined above, three kind of graphs, or networks, are used to represent strength of associations among Web pages: (1) the hypetext link topology of a Web locality, (2) interpage text similarity, and (3) the usage paths, or flow of users through the locality. Each of these networks or graphs is represented by matrices in our spreading activation algorithm. That is, each row corresponds to a network node representing a Web page, and similarly each column corresponds to a network node representing a Web page. If we index the 1, 2, ... N Web pages, there would be i=1, 2, . . . N columns and j=1, 2, . . . N rows for each matrix representing a graph network.

Each entry in the i^{th} column and j^{th} row of a matrix represents the strength of connection between page i and page j (or similarly, the amount of potential activation flow or capacity). The meaning of these entries varies depending on the type of network through which activation is being spread.

FIGS. 8–9 illustrate a topology network for a Web locality and the corresponding matrix representation. Referring to FIG. 8, each node or Web page is represented as feature vectors 801–807. The arcs in the graph indicate links between the various pages. Referring now to FIG. 9, for the matrix representation in topology networks, an entry of 0 in column i, row j, indicates no hypertext link between page i and page j, whereas an entry of 1 indicates a hypertext link. So for example, Web page 801 is seen to have links to pages 802–804 by the entry of 1 in the corresponding positions of the topology matrix.

FIGS. 10–11 illustrate a text similarity network and corresponding matrix representation. Referring to FIG. 10, the widths of the lines connecting the various pages 1001–1007 is an indication of how similar the pages are. Referring now to FIG. 11, for the matrix representation of text similarity networks, an entry of a real number, s>=0, in column i, row j indicates the inter-document similarity of page i to page j.

VARYING WIDTHS OF LINES

FIGS. 12–13 illustrate usage path network and corresponding matrix representation. Referring to FIG. 12, it should be noted that there will only be usage between nodes where there are corresponding links. So illustrated on FIG. 12 are both links and the usage path. Referring now to FIG. 13, for the matrix representation of usage path networks, an entry of an integer strength, s>=0, in column i row j, indicates the number of users that traversed from page i to page j.

Activation-

An activation network can be represented as a graph defined by matrix R, where each off-diagonal element Ri,j contains the strength of association between nodes i and j, and the diagonal contains zeros. The strengths determine how much activation flows from node to node. The set of source nodes of activation being pumped into the network is represented by a vector C, where Ci represents the activation pumped in by node i. The dynamics of activation can be modeled over discrete steps t=1, 2, . . . N, with activation at step t represented by a vector A(t), with element A(t, i) representing the activation at node i at step t. The time evolution of the flow of activation is termined by

$$A(t)=C+M\,A(t-1) \qquad \text{Equation 2}$$

where M is a matrix that determines the flow and decay of activation among nodes. It is specified by

$$M=(1-g)I+\alpha R, \qquad \text{Equation 3}$$

where g<1 is a parameter determining the relaxation of node activity back to zero when it receives no additional activation input, and a is a parameter denoting the amount of activation spread from a node to its neighbors. I is the identity matrix.

Example 1

Predicting the Interests of Home Page Visitors

To illustrate, consider the situation in which it is desirable to identify the most frequently visited organization home page using the categorization information, and construct a Web aggregate that contains the pages most visited from that page. The most popular organization page can be identified by first finding the pages in that category using the classification criteria described in FIG. 6 (i.e. the "Organization Home Page" criteria). The most popular page would then be the identified page having the highest "frequency" value in their corresponding document vector. To find the most visited page through spreading activation, the corresponding component of C given a positive value, and the remaining elements set to zero. Setting the association matrix R to be the usage path matrix, Equation 2 above is iterated for N time steps (e.g. N=10 has provided acceptable results). The most visited pages are then those having the highest activation. Alternatively, the most visited pages may be those that exceed some predetermined activation threshold. In any event, a Web aggregate has been identified.

Based on this information traversal patterns can be determined which identify the most popular types of information requested. So an external user entering a companies home page may be looking at the companies products or financial reports. This may give a profile that the typical person examining the Web locality are potential customers or investors.

Example 2

Assessing the Typical Web Author at a Locality

Consider another situation in which the Web pages of interest are those having the highest text similarity to the most typical person page in a Web locality. In other words, one might be interested in understanding something about what a typical person publishing in a Web locality says about themselves. In this case, the most typical person page is identified using the "Personal Home Page" Criteria described in FIG. 6, the corresponding C element set to positive activation input (zeros elsewhere), and R is set to the text similarity matrix. Iteration of this spread of activation for N=10 time steps selects a collection of Web pages. By reading the group project overviews, the home pages of related people, personal interest pages, and formal and informal groups to which the person belongs, one may get some sense of what people are like in the organization.

Combining Activation Nets

Because of the simple properties of the activation networks, it is easy to combine the spread of activation though any weighted combination of activation pumped from different sources and through different kinds of arc—that is, simultaneously through the topology, usage, and text similarity connections. Consequently, the Web locality can be lit up from different directions and using different colors of predicted relevancy. For instance one might be interested in the identifying the pages most similar in content to the pages most popularly traversed.

Visualizations-

Most current Web browsers provide very little support for helping people gain an overall assessment of the structure and content of large collections of Web pages. Information Visualization could be used to provide an interactive overview of web localities that facilitates navigation and general assessment. Visualizations have been developed that provide new interactive mechanisms for making sense of information sets with thousands of objects. The general approach is to map properties and relations of large collections of objects onto visual, interactive structures.

To the extent that the properties that help users navigate around the space and remember locations or ones that support the unit tasks of the user's work, the visualizations provide value to the user. Visualizations can be applied to the Web by treating the pages of the Web as objects with properties. Each of these visualizations provide an overview of a Web locality in terms of some simple property of the pages. For example, the present invention may be used in support of information visualization techniques, such as the WebBook described in co-pending and commonly assigned

WEBBOOK

application Ser. No. 08/525,936 entitled "Display System For Displaying Lists of Linked Documents", to form and present larger aggregates of related Web pages. Other examples include a Cone Tree which shows the connectivity structure between pages and a Perspective Wall which shows time-indexed accesses of the pages. The cone tree is described in U.S. Pat. No. 5,295,243 entitled "Display of Hierarchical Three-Dimensional Structures With Rotating Substructures". The Perspective Wall is described in U.S. Pat. No. 5,339,390 entitled "Operating A Processor To Display Stretched Continuation Of A Workspace". Thus, these visualizations are based on one or a few characteristics of the pages.

Overview of a Computer Controlled Display System in the Currently Preferred Embodiment of the Present Invention

The computer based system on which the currently preferred embodiment of the present invention may be implemented is described with reference to FIG. 14. The computer based system and associated operating instructions (e.g. software) embody circuitry used to implement the present invention. Referring to FIG. 14, the computer based system is comprised of a plurality of components coupled via a bus **1401**. The bus **1401** may consist of a plurality of parallel buses (e.g. address, data and status buses) as well as a hierarchy of buses (e.g. a processor bus, a local bus and an I/O bus). In any event, the computer system is further comprised of a processor **1402** for executing instructions provided via bus **1401** from Internal memory **1403** (note that the Internal memory **1403** is typically a combination of Random Access and Read Only Memories). The processor **1402** will be used to perform various operations in support extracting raw data from web localities, converting the raw data into the desired feature vectors and topology, usage path and text similarity matrices, categorization and spreading activation. Instructions for performing such operations are retrieved from Internal memory **1403**. Such operations that would be performed by the processor **1402** would include the processing steps described in FIGS. 1–4 and 7. The operations would typically be provided in the form of coded instructions in a suitable programming language using wellknown programming techniques. The processor **1402** and Internal memory **1403** may be discrete components or a single integrated device such as an Application Specification Integrated Circuit (ASIC) chip.

Also coupled to the bus **1401** are a keyboard **1404** for entering -alphanumeric input, external storage **1405** for storing data, a cursor control device **1406** for manipulating a cursor, a display **1407** for displaying visual output (e.g. the WebBook) and a network connection **1408**. The keyboard **1404** would typically be a standard QWERTY keyboard but may also be telephone like keypad. The external storage **1405** may be fixed or removable magnetic or optical disk drive. The cursor control device **1406**, e.g. a mouse or trackball, will typically have a button or switch associated with it to which the performance of certain functions can be programmed. The network connection **1408** provides means for attaching to a network, e.g. a Local Area Network (LAN) card or modem card with appropriate software. The network ultimately attached to is the Internet, but it may be through intermediary networks or On-Line services such as America On-Line, Prodigy ™ or CompuServ ™.

Thus, a system for analyzing a collection of hyper-linked pages is disclosed. While the present invention is described with respect to a preferred embodiment, it would be apparent to one skilled in the art to practice the present invention with other configurations of digital document management systems. Such alternate embodiments would not cause departure from the spirit and scope of the present invention. For example, the present invention may be implemented as software instructions residing on a suitable memory medium for use in operating a computer based system.

What is claimed is:

1. A system for categorizing documents contained in a linked collection of documents comprising:

means for obtaining raw data from said linked collection of documents, said raw data including meta information for documents in said linked collection of documents;

means for creating a feature vector for documents in said linked collection of documents from said raw data, said feature vector comprising a plurality of elements;

means for defining classification criteria indicating particular categories of document types, said classification criteria comprising user defined weightings of the elements for said feature vector and a corresponding class threshold value;

processing means for applying said classification criteria to feature vectors to determine if a document is in a corresponding category.

2. The system as recited in claim 1 wherein said means for obtaining raw data for said linked collection of documents is further comprised of a first agent for traversing said linked collection of documents to obtain topology information and document meta information.

3. The system as recited in claim 2 wherein the plurality of elements of a feature vector for a document in said linked collection of documents include:

size information for said document;

inlink information for said document, said inlink information indicating the number of links in said linked collection of documents that point to said document;

outlink information for said document, said outlink information indicating the number of links the document contains to other documents said linked collection of documents;

frequency information for said document, said frequency information indicating the number of times said document was requested during a sample period;

source information for said document, said source information indicating the number of times said document was identified as the start of a path traversal;

text similarity information for said document, said text similarity information indicating the similarity of the text of the document to documents in said linked collection of document to which they are linked; and

depth information for said document, said depth information indicating the average depth in said linked collection of documents of documents to which said document links.

4. The system as recited in claim 3 wherein said processing is comprised of means for determining that a document is a class if after applying said classification criteria the result exceeds said corresponding class threshold value.

5. The system as recited in claim 1 wherein said linked collection of documents is a Web locality.

6. A method for generating a list of web pages in a web locality that are contained in a user defined class comprising the steps of:

a) obtaining raw data for said web locality, said raw data including topology information and web locality usage information;

b) generating page meta data for each web page in said web locality from said raw data;

c) generating feature vectors for each web page in said web locality using said page meta data and said topology information, said feature vector comprised of a plurality of elements;

d) obtaining a classification criteria for determining if a web page is a member of a category of web pages, said classification criteria comprising user defined weightings of the plurality of elements for said feature vector and a corresponding class threshold value; and

e) applying said classification criteria to said feature vectors to obtain a list of pages in said category.

7. The method as recited in claim 6 wherein said step of obtaining topology information for said web locality is comprised of the steps of:

a1) retrieving a web page;

a2) storing location information for said web page;

a3) parsing said web page to identify links to other web pages; and

a4) repeating steps a1)–a3) for each of said other web pages.

8. The method as recited in claim 6 wherein said step of obtaining page meta data for each web page in said web locality is further comprised of the step of collecting page meta data for a page as the page is retrieved.

9. The method as recited in claim 6 wherein said step of generating feature vectors for each web page in said web locality using said page meta data and said topology information is further comprised of the step of for each associated web page in said web locality performing the steps of:

extracting size information for said associated web page and storing as a size element in said corresponding feature vector;

extracting inlink information for said associated web page, said inlink information indicating the number of links in said web locality that point to said associated web page as storing as an inlink element in said corresponding feature vector;

extracting outlink information for said associated web page, said outlink information indicating the number of links the web page contains to other web pages in said web locality and storing as an outlink element in said corresponding feature vector;

extracting frequency information for said associated web page, said frequency information indicating the number of times said associated web page was requested during a sample period and storing as a frequency element in said corresponding feature vector;

extracting source information for said associated web page, said source information indicating the number of times said associated web page was identified as the start of a path traversal and storing as a source element in said corresponding feature vector;

extracting text similarity information for said associated web page, said itext similarity information indicating the similarity of the text of the associated web page to other web pages in said web locality to which they are linked and storing as a text similarity element in said corresponding feature vector; and

extracting depth information for said associated web page, said depth information indicating the average depth in said web locality of other web pages to which said associated web page links and storing as a depth element in said corresponding feature vector.

10. The method as recited in claim 9 wherein said step of applying said classification criteria to said feature vectors to obtain a list of pages in said category is further comprised of the steps of:

for each element of a feature vector applying a corresponding weighting value to obtain a feature value;

summing the resulting values feature values; and

comparing said sum to said class threshold value to determine if said corresponding page is in said class.

11. A system for generating characteristic data for a linked collection of documents comprising:

means for obtaining raw data for said linked collection of documents, said raw data including usage data, topology data and content data;

means for creating a feature vector for each document in said linked collection of documents from said raw data; and

means for categorizing each of said documents in said linked collection of documents according to predetermined classification criteria, said predetermined classification criteria comprising user defined weightings of the elements for said feature vector and a corresponding class threshold value.

12. The system as recited in claim 11 further comprising:

means for creating usage, topology and text similarity maps for said linked collection of documents from said raw data;

means for predicting a relevant set of documents for a subset of said linked collection of documents using one or more of said usage, topology and text similarity maps.

13. A system for categorizing documents contained in a linked collection of documents comprising:

means for obtaining raw data from said linked collection of documents, said raw data including meta information for documents in said linked collection of documents;

means for creating a feature vector for documents in said linked collection of documents from said raw data, said feature vector having at least one element indicating a frequency of request for an associated document;

means for defining classification criteria indicating particular categories of document types;

processing means for applying said classification criteria to feature vectors to determine if a document is in a corresponding category.

14. A method for generating a list of web pages in a web locality that are contained in a user defined class comprising the steps of:

a) obtaining raw data for said web locality, said raw data including topology information and web locality usage information;

b) generating page meta data for each web page in said web locality from said raw data, said meta data including data indicating a frequency of request for an associated document;

c) generating feature vectors for each web page in said web locality using said page meta data and said topology information;

d) obtaining a classification criteria for determining if a web page is a member of a category of web pages; and

e) applying said classification criteria to said feature vectors to obtain a list of pages in said category.

* * * * *